REMARKS

Claims 1-30, as amended, remain herein.

Editorial changes have been made in claims 1-30.

The specification has been edited to replace "selector 124" with "register 113" and "selector 24" with "register 13" in numerous paragraphs, and other edits for clarity, consistent with Figs. 1 and 7. Also, the specification has been amended to note the mathematical notation employed therein, i.e., the formula "LxM-1" herein means "(LxM)-1", e.g., (4x5)-1= 19. See specification, page 48, line 10.

Submitted herewith are copies of Figures 1 and 7 revised to replace a line connecting element 121 to a connection point on a line between elements 124 and 126 with a line connecting element 121 to a connection point on a line between elements 113 and 128. Figures 1 and 7 have been further revised to express the formula as "(LxM)-1".

A verified English-language translation of the priority application Japanese Patent Application No. 9-360863 has been submitted herewith.

1. The title has been amended to read "Block Interleaving Apparatus, Block Deinterleaving Apparatus, Block Interleaving Method and Block Deinterleaving Method."

2. Applicant acknowledges on the record the July 30, 2004 telephone interview between applicant's representative, Robert N. Wieland, and Examiner Dipakkumar Gandhi. During that interview, the Examiner confirmed that the foreign priority claim filed on October 5, 2001 has been entered, as later stated in the Interview Summary mailed August 9, 2004.

3. Claims 1-9 and 19-24 were rejected under 35 U.S.C. §103(a) over Ito et al. Japanese Patent Application 2000078030 and Min U.S. Patent 5,430,767.

The present application includes a claim for foreign priority effective March 15, 1999 under 35 U.S.C. §119 as stated in applicants' Declaration and Power of Attorney, filed October 5, 2001. In the verified English-language translation of the priority application submitted herewith, paragraphs 0038, 0095, 00127, and 0144 support claims 1, 2 and 19, and thus antedate Ito JP '030. Ito JP '030 was published on March 14, 2000, and therefore antedates applicant's foreign priority date of March 15, 1999. Accordingly, the combination of Min '767 with Ito JP '030 is improper in the present rejection.

Moreover, Min '767 does not provide the structure alleged in the Office Action as explained next:

The presently claimed block interleaving apparatus includes a storage means for data writing and data reading, an address generation means for generating addresses for writing and reading blocks, and a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means; the address generation means including a multiplication means for generating a product, a first overflow processing means, an addition means, and a second overflow processing means, wherein, the first comparison means is for comparing the product obtained by the multiplication with a comparison reference value (L×M)-1, the first comparison means employs, as a comparison reference value instead of the value (L×M)-1, the minimum value "A" which exceeds the value (L×M)-1 and is included in the product, as recited in claims 1, 2, 10, 11 and

19. This arrangement is nowhere disclosed or suggested in either of the cited references.

The block interleaving apparatus recited in claims 1, 2, 10, 11 and 19 compares the values by employing, as a comparison reference value instead of(L×M)-1, the minimum value "A" which exceeds the value (L×M)-1 in the comparator of the modulo operation device in the address generation means in order to reduce the circuit area of the comparator by changing the address initial value alpha (alpha being an integer ≥2, in the address generation means according to the values of L and M in a unit of (LxM) pieces of data to be subjected to block interleaving. Thus the circuit area and the calculation amount of the modulo operation device in the address generation device are reduced. As a result, effects such as a decrease in electric power consumption and reduction in the area are obtained.

The Office Action cites Min '767 as allegedly disclosing a block interleaving apparatus and control means for controlling the storage means to switch the operation between the data writing and the data reading by using the addresses generated by the address generation means. Min '767 discloses two memory devices, and therefore, its construction is totally different from that of the presently claimed invention in which block interleaving is performed in one memory device (i.e., "a storage means").

Min '767 does not disclose or suggest a block interleaving apparatus that compares the values by employing, as a comparison reference value instead of (L×M)-1, the minimum value "A" which exceeds the (L×M)-1 in the comparator of the modulo operation device in the address generation means in order to reduce the circuit area of the comparator by changing the address initial value alpha, wherein alpha is an integer ≥2, in the address generation means according to the values of L and M in a unit of (LxM) pieces of data to be subjected to block interleaving, as recited in applicant's claims 1, 2, 10, 11 and 19.

For the foregoing reasons, Min '767 does not contain any teaching, suggestion, reason, motivation or incentive that would have led one of ordinary skill in the art to applicant's claimed invention. Ito JP '030 is not applicable, as explained herein. Claims 3-9, which depend from claim 2, are allowable for the same reasons explained herein for claim 2, and claims and 20-24, which depend from claim 19, are allowable for the same reasons explained herein for claim 19.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

4. Claims 10-18 and 25-30 were rejected under 35 U.S.C. §103(a) over Ito '030, Min '767 and Horii et al. JP 20001036765.

The present application includes a claim for foreign priority effective March 15, 1999 under 35 U.S.C. §119 as stated in applicant's Declaration and Power of Attorney, filed October 5, 2001. In the verified English translation of the priority application submitted herewith, paragraphs 0135 and 0096-0099 support claims 10 and 23 and thus antedate Horii JP '765. Ito JP '030 is invalid as a proper reference as explained herein. Horii JP '765 was published on February 2, 2000, and therefore antedates applicant's foreign priority date of March 15, 1999. Accordingly, the combination of Horii JP '765 and Ito JP '030 is <u>improper</u> in the present rejection, and reconsideration and withdrawal of the rejection are respectfully requested.

All claims 1-30 are now proper in form and patentably distinguished over all grounds of rejection stated in the Office Action. Accordingly, allowance of all claims 1-30 is respectfully requested.

Should the Examiner deem that any further action by the applicant would be desirable to place this application in even better condition for issue, the Examiner is requested to telephone applicant's undersigned representatives.

Respectfully submitted,

PARKHURST & WENDEL, L.L.P.

November 12, 2004
Date

Roger W. Parkhurst
Registration No. 25,177
Robert N. Wieland
Registration No. 40,225

RWP:RNW/mhs

<u>Attachments</u>: 2 annotated sheets showing changes - Figs. 1 & 7
2 replacement sheets - Figs. 1 & 7
Verified English-Language Translation of
JP Patent Application No. 9-360863

Attorney Docket No.: <u>HYAE:127</u>

PARKHURST & WENDEL, L.L.P.
1421 Prince Street, Suite 210
Alexandria, Virginia 22314-2805
Telephone: (703) 739-0220

ANNOTATED SHEET SHOWING CHANGES
SERIAL NO.: 09/936,510
CONFIRMATION NO.: 4256
Senichi FURUTANI
FILED: October 5, 2001

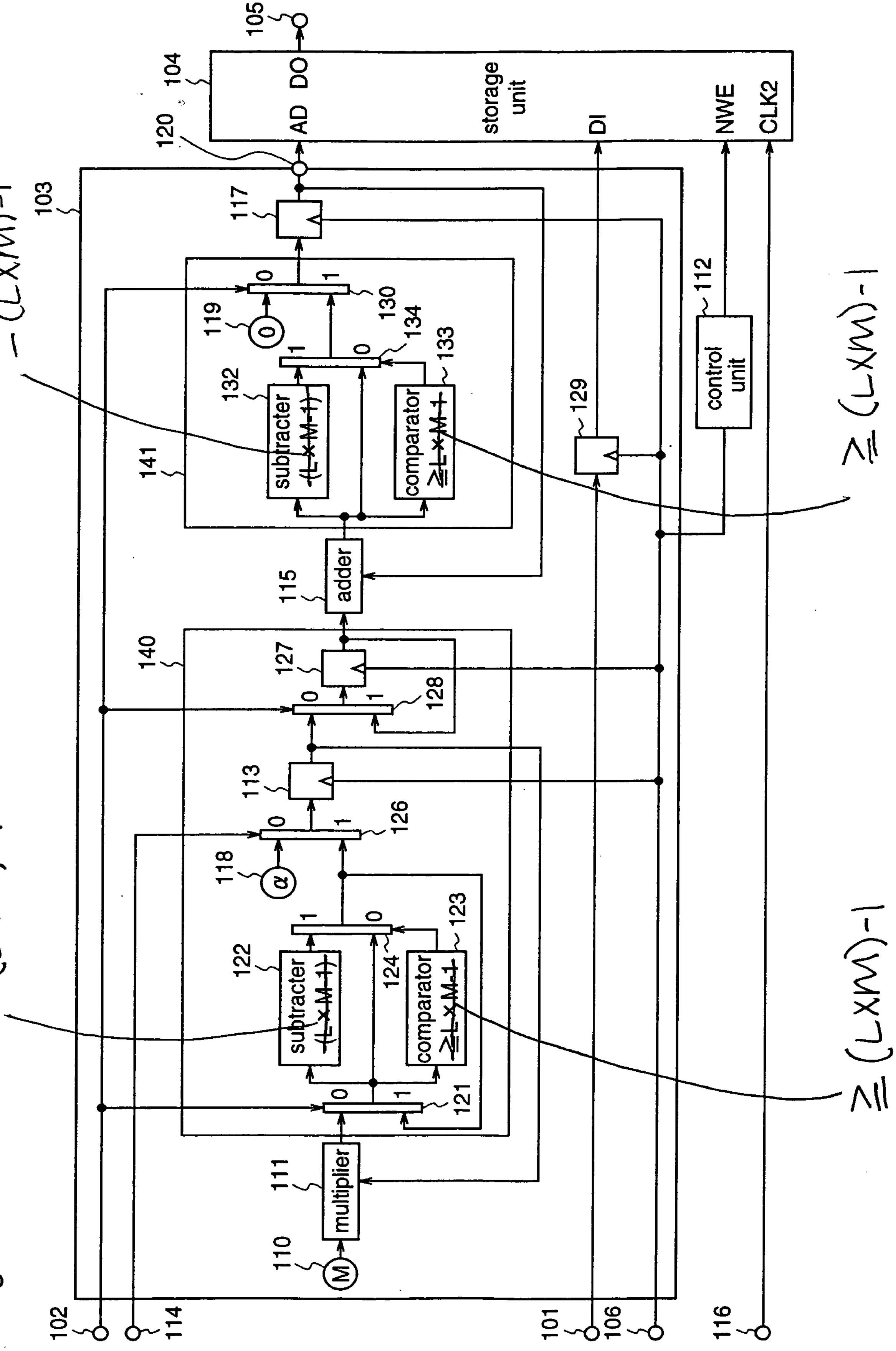


Fig.1

ANNOTATED SHEET SHOWING CHANGES
SERIAL NO.: 09/936,510
CONFIRMATION NO.: 4256
Senichi FURUTANI
FILED: October 5, 2001

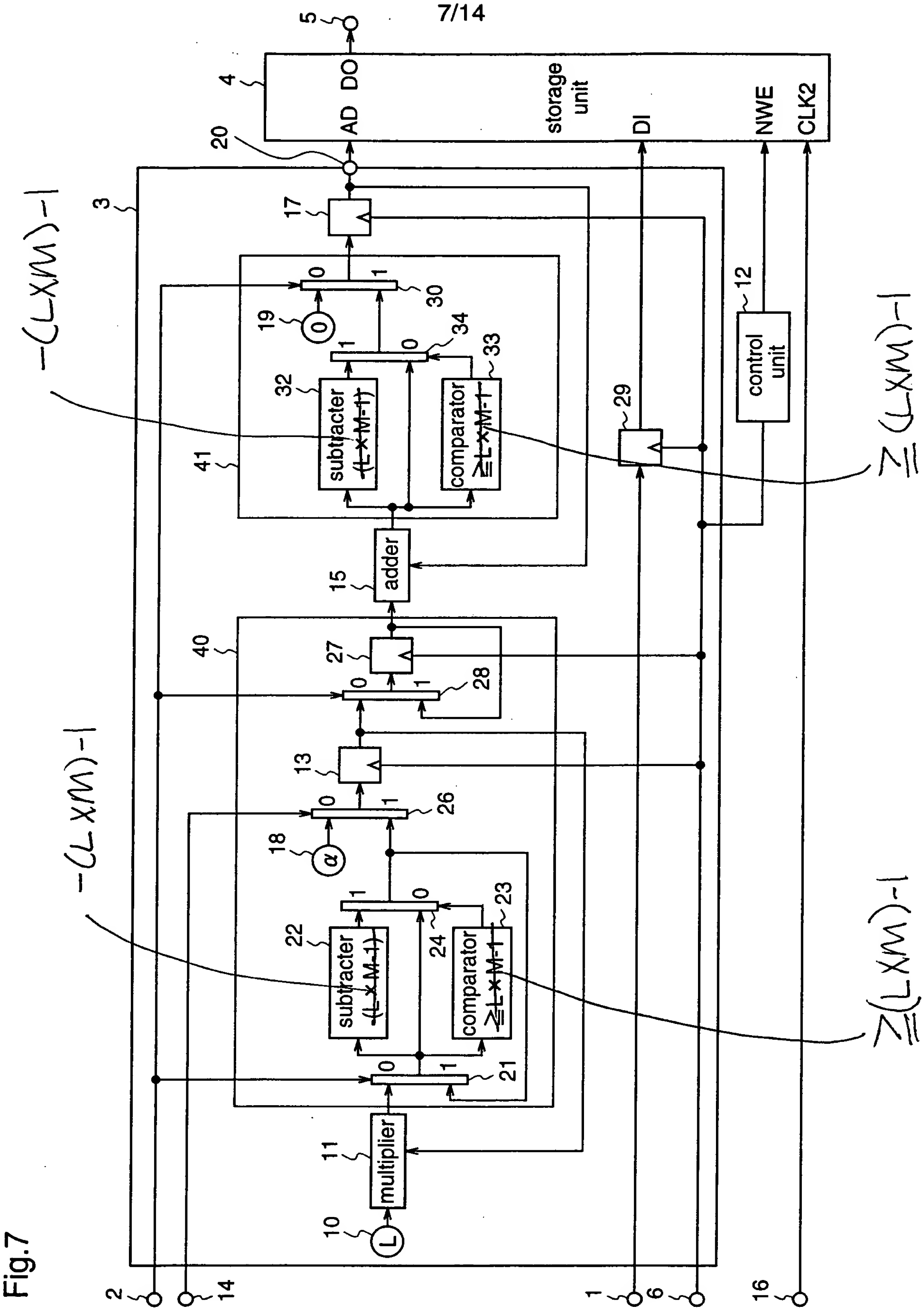


Fig.7

Certificate

I, Yuko Yagi, a member of Hayase & Company Patent Attorneys of 13F, NISSAY SHIN-OSAKA Bldg., 3-4-30, Miyahara, Yodogawa-ku, Osaka-shi, Osaka 532-0003 Japan, hereby certify that to the best of my knowledge and belief the following is a true translation into English made by me of the Japanese Patent Application Number 9-360863.

Osaka, this 9th day of November, 2004

Yuko Yagi

Yuko Yagi

[Name of Document] Patent Application

[Reference Number] 2037810014

[Filing Date] March 15, 1999

[Destination] Commissioner, Patent Office

[International Patent Classification] H03M 13/22

[Title of Invention] BLOCK INTERLEAVING APPARATUS, BLOCK DEINTERLEAVING APPARATUS, BLOCK INTERLEAVING METHOD, AND BLOCK DEINTERLEAVING METHOD

[Number of Claims] 20

[Inventor]

[Address] c/o Matsushita Soft-Research, Inc.
1006 Oaza Kadoma Kadoma-shi,
Osaka JAPAN

[Name] Senichi Furutani

[Applicant]

[Identification No.] 000005821

[Name] Matsushita Electric Industrial Co., Ltd.

[Patent Attorney]

[Identification No.] 100081813

[Patent Attorney]

[Name] Kenichi Hayase

[Telephone Number] 06(6380)5822

[Representation of Fee]

[Ledger No.] 013527

[Amount of Payment] 21,000

[Attached Documents]

[Name of Document] Specification 1

[Name of Document] Drawing 1

[Name of Document] Abstract 1

[Number of General Authorization] 9600402

[Name of the Document] Specification

[Title of the Invention] Block Interleaving Apparatus, Block Deinterleaving Apparatus, Block Interleaving Method, and Block Deinterleaving method

[Claims]

[Claim 1] A block interleaving apparatus characterized by:

performing writing while skipping addresses at the same interval corresponding to at least one address, into a storage unit to which the addresses are sequentially assigned;

performing reading in such order that the read data become block-interleaved data to obtain output data; and

thereafter, performing writing in the same order as the reading, thereby performing block interleaving.

[Claim 2] A block interleaving apparatus as defined in Claim 1 comprising:

a storage unit for L×M pieces of data for generating an output of the block interleaving apparatus;

an address generation unit for generating addresses to be output to the storage unit; and

a storage control unit for outputting a control signal to the storage unit;

wherein, in said address generation unit, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $M^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and

reading and writing are repeated on the generated addresses to perform block interleaving.

[Claim 3] A block interleaving apparatus as defined in Claim 2, wherein the values of α and L×M-1 are determined so that no common divisor exists between them.

[Claim 4] A block interleaving apparatus as defined in Claim 2, wherein the values of α and $M^{(-x)}$ are determined so that α is not equal to $M^{(-x)}$.

[Claim 5] A block interleaving apparatus as defined in Claim 2, wherein the values of α, L, and M are determined at 20, 8, and 203, respectively.

[Claim 6] A block deinterleaving apparatus characterized by:

performing writing while skipping addresses at the same interval corresponding to at least one address, into a storage unit to which the addresses are sequentially assigned;

performing reading in such order that the read data become block-deinterleaved data to obtain output data; and

thereafter, performing writing in the same order as the reading, thereby performing block deinterleaving.

[Claim 7] A block deinterleaving apparatus as defined in Claim 6 comprising:

a storage unit for L×M pieces of data for generating an output of the block deinterleaving apparatus;

an address generation unit for generating addresses to be output to the storage unit; and

a storage control unit for outputting a control signal to the storage unit;

wherein, in said address generation unit, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $L^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and

reading and writing are repeated on the generated addresses to perform block interleaving.

[Claim 8] A block deinterleaving apparatus as defined in Claim 7, wherein the values of α and L×M-1 are determined so that no common divisor exists between them.

[Claim 9] A block deinterleaving apparatus as defined in Claim 7, wherein the values of α and $L^{(-x)}$ are determined so that α is not equal to $L^{(-x)}$.

[Claim 10] A block deinterleaving apparatus as defined in Claim 7, wherein the values of α, L, and M are determined at 20, 8, and 203, respectively.

[Claim 11] A block interleaving method characterized by:

performing writing while skipping addresses at the same interval corresponding to at least one address, into a storage unit to which the addresses are sequentially assigned;

performing reading in such order that the read data become block-interleaved data to obtain output data; and

thereafter, performing writing in the same order as the reading, thereby performing block interleaving.

[Claim 12] A block interleaving method as defined in Claim 11, wherein 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $M^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving.

[Claim 13] A block interleaving method as defined in Claim 12, wherein the values of α and L×M-1 are determined so that no common divisor exists between them.

[Claim 14] A block interleaving method as defined in Claim 12, wherein the values of α and $M^{(-x)}$ are determined so that α is not equal to $M^{(-x)}$.

[Claim 15] A block interleaving method as defined in Claim 12, wherein the values of α, L, and M are determined at 20, 8, and 203, respectively.

[Claim 16] A block deinterleaving method characterized by:

performing writing while skipping addresses at the same interval corresponding to at least one address, into a storage unit to which the addresses are sequentially assigned;

performing reading in such order that the read data become block-deinterleaved data to obtain output data; and

thereafter, performing writing in the same order as the reading, thereby performing block deinterleaving.

[Claim 17] A block deinterleaving method as defined in Claim 16, wherein 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $L^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving.

[Claim 18] A block deinterleaving method as defined in Claim 17, wherein the values of α and L×M-1 are determined so that no common divisor exists between them.

[Claim 19] A block deinterleaving method as defined in Claim 17, wherein the values of α and $L^{(-x)}$ are determined so that α is not equal to $L^{(-x)}$.

[Claim 20] A block deinterleaving method as defined in Claim 17, wherein the values of α, L, and M are determined at 20, 8, and 203, respectively.

[Detailed Description of the Invention]

[0001]

[Technical Filed of the Invention]

The present invention relates to a block interleaving apparatus, a block deinterleaving apparatus, a block interleaving method, and a block deinterleaving method, which are required for digital transmission such as satellite broadcasting, ground wave broadcasting, or cable broadcasting, and for reading and writing of a storage unit such as a hard disk.

[0002]

[Prior Art]

Block interleaving is effective as a countermeasure against burst errors.

[0003]

Hereinafter, block interleaving will be described taking satellite broadcasting as an example. A radio wave from a broadcast station on earth is transmitted to a satellite, relayed by the satellite, and received by a satellite broadcast receiver provided at home.

[0004]

The radio wave, which is transmitted from the broadcast station through the satellite to home, might be subjected to interference by thunder, rain or the like in the transmission path. While the radio wave is subjected to such interference, errors occur in data. These errors are called "burst errors".

[0005]

In digital transmission, since information for error correction has already been added to the original data, errors can be corrected so long as the errors are within a predetermined number of bits in each segment. However, continuous errors such as burst errors cannot be corrected.

[0006]

So, data to be transmitted is temporally dispersed in advance (a method for this data dispersion is block interleaving), whereby, even if burst errors occur during transmission, these burst errors are also dispersed when the temporal positions of the dispersed data are recovered at the receiving end (a method for this recovery is block deinterleaving) and, in each data block, the burst errors can be limited within a number of bits which can be corrected.

[0007]

When performing such block interleaving and block deinterleaving, two planes of storage units, each having a storage area of 1 block (L×M data) originally, are required, and writing and reading are alternately repeated on these storage units. Japanese Published Patent Application No. Hei.8-511393 discloses block interleaving and block deinterleaving which can be realized with reduced circuit scale and reduced power consumption.

[0008]

Figure 13 is a diagram schematically illustrating the operation of the conventional block interleaving, wherein, for simplification, block interleaving is performed on 4 rows × 5 columns of data.

Assuming that addresses of a storage unit of a block interleaving apparatus are allocated as shown in figure 13(a), initially, REG is set at 1, and data are sequentially written in the order of 0 → 1 → 2 → .... → 19, i.e., in order as the address increments. Next, as shown in figure 13(b), data are read out in the order as the address increments. That is, the REG is multiplied by 5, and an address which increments by 5 at every data input is successively generated with address 0 shown in figure 13(a) being an initial value. At this time, when the address exceeds 19 (=4×5-1), this 19 is subtracted. Then, according to the addresses generated under the generation rule, initially, the data which have already been written as shown in figure 13(a) are read out in the order of the generated addresses and, after the readout is completed, writing of data is carried out in the same order as that for the data reading.

[0009]

Next, as shown in figure 13(c), the REG is multiplied by 5, and when the value exceeds 19, this 19 is subtracted.

Then, using address 0 as an initial value with respect to the arrangement of addresses shown in figure 13(a), an address which increments by 6 for every input data is successively generated and, when the address exceeds 19 (=4×5-1), reading is carried out by subtracting this 19. After the reading is completed, data are written in the same order as that for the reading.

[0010]

Thereafter, by repeating the same process as described above, reading and writing are carried out in different address orders, whereby, in this example, the address order returns to that shown in figure 13(a) at the point of time shown in figure 13(j).

By repeating the above-described procedure, it is possible to perform block interleaving using a RAM 202 having a storage area of one block (L×M data), as shown in figure 14. The block interleaving is realized by contriving, as described above, the writing/reading control by the RAM control apparatus 200 and the addresses generated by the address generation unit 201.

[0011]

The address generation rule employed in the conventional block interleaving apparatus is as follows.

That is, assuming that the n-th address is Ab(n), the number of rows of the storage unit is L, the number of columns is M, b is an integer not less than 0, and x is an arbitrary integer not less than 0 and not larger than b,

$$Ab(n)=(Ab(n-1)+M^{**}(b-x))\bmod(L\times M-1) \qquad \ldots(1)$$

Further,

$$REG=(M^{**}(b-x))\bmod(L\times M-1)$$

wherein Ab(0) is 0, and M**(b-x) indicates the (b-x)th power of M.

[0012]

Further, block deinterleaving is performed as follows on the data which have been subjected to the above-described block interleaving. Assuming that addresses of a storage unit of a block deinterleaving apparatus are allocated as shown in figure 13(k), initially, the REG is set at 1, and data are sequentially written in the addresses in the order of 0 → 1 → 2 → ... → 19, i.e., according to increment of the addresses. Next, as shown in figure 13(l), the data are read out. That is, the REG is multiplied by 4, and an address which increases by 4 for every input data is sequentially generated with address 0 shown in figure 13(k) being an initial value. At this time, when the address exceeds 19 (=4×5-1), this 19 is subtracted. Then, according to the addresses generated under this address generation rule, initially, the data which have already been written as shown in figure 13(k) are sequentially read out in the order of the generated addresses. After the readout has been completed, the data are written in the same order as those for the readout.

[0013]

Next, as shown in figure 13(m), the REG is multiplied by 4, and when the value exceeds 19, this 19 is subtracted.

Then, an address which increments by 16 for every input data is sequentially generated by using address 0 as an initial value with respect to the arrangement of addresses shown in figure 13(k), and when the address exceeds 19 (=4×5-1), reading is carried out by subtracting this 19. After the reading has been completed in figure 13(m), data are written in the same order as those for the reading.

[0014]

By repeating the same process as above, reading and writing are sequentially carried out in different address orders, whereby the address order returns to that shown in figure 13(k) at the point of time shown in figure 13(t).

By repeating the above-described procedure, it is possible to perform block deinterleaving by using the RAM 202 having a storage area of one block, as shown in figure 14. This block deinterleaving is realized by contriving the writing/reading control by the RAM control apparatus 200 and the addresses generated by the address generation unit 201 as described above.

[0015]

The address generation rule employed in the conventional block deinterleaving apparatus is as follows.

$$Ab(n)=(Ab(n-1)+L^{**}(b-x))\bmod(L\times M-1) \qquad \ldots(2)$$

Further,

$$REG=(L^{**}(b-x))\bmod(L\times M-1)$$

wherein Ab(0) is 0.

In formula (2), M in formula (1) is changed to L.

[0016]

[Problems to be solved by the Invention]

The conventional block interleaving apparatus and block deinterleaving apparatus are constructed as described above, and these apparatuses can perform block interleaving and block deinterleaving by using only one storage unit having a storage area corresponding to one block, whereby reduced circuit scale and reduced power consumption are realized.

However, as the result of an earnest research by the inventors of the present invention, it turned out that this conventional processing system has the scope for further reduction in the circuit area and reduction in power consumption.

[0017]

The present invention has for its object to provide a block interleaving apparatus, a block deinterleaving apparatus, a block interleaving method, and the block deinterleaving method, which can realize further reduction in circuit scale and further reduction in power consumption, by optimizing control units for storage units.

[0018]

[Measures to solve the Problems]

According to the invention of Claim 1, a block interleaving apparatus performs writing while skipping addresses at the same interval corresponding to at least one address into a storage unit to which the addresses are sequentially assigned, and performs reading in such order that the read data become block-interleaved data to obtain output data, and thereafter, performs writing in the same order as the reading, thereby performing block interleaving.

Since the block interleaving apparatus according to the invention of Claim 1 performs the above-mentioned writing and reading operations, it enables a block interleaving operation on a single plane of a storage unit having a storage area of one block, and reduces the circuit scale of a storage control unit.

[0019]

According to the invention of Claim 2, the block interleaving apparatus defied in Claim 1 comprises a storage unit for L×M pieces of data for generating an output of the block interleaving apparatus; an address generation unit for generating addresses to be output to the storage unit; and a storage control unit for outputting a control signal to the storage unit; wherein, in said address generation unit, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $M^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving.

Since the block interleaving apparatus according to the invention of Claim 2 is constituted as described above, it optimizes the storage unit and the address generation unit, and enables block interleaving with a minimum circuit area.

[0020]

According to the invention of Claim 3, in the block interleaving apparatus defined in Claim 2, the values of α and L ×M-1 are determined so that no common divisor exists between them.

Since the block interleaving apparatus according to the invention of Claim 3 is constituted as described above, it prevents the address generation rule from being failed, optimizes the storage unit and the address generation unit, and enables block interleaving with a minimum circuit area.

[0021]

According to the invention of Claim 4, in the block interleaving apparatus defined in Claim 2, the values of α and $M^{(-x)}$ are determined so that α is not equal to $M^{(-x)}$.

Since the block interleaving apparatus according to the invention of Claim 4 is constituted as described above, it prevents writing into continuous addresses at the initial writing, optimizes the storage unit and the address generation unit, and enables block interleaving with a minimum circuit area.

[0022]

According to the invention of Claim 5, in the block interleaving apparatus defined in Claim 2, the values of α, L, and M are determined at 20, 8, and 203, respectively.

Since the block interleaving apparatus according to the invention of Claim 5 is constituted as described above, it reduces the area of a comparator as a component of the address generation unit, optimizes the storage unit and the address generation unit, and enables block interleaving with a minimum circuit area.

[0023]

According to the invention of Claim 6, a block deinterleaving apparatus performs writing while skipping addresses at the same interval corresponding to at least one address into a storage unit to which the addresses are sequentially assigned, and performs reading in such order that the read data become block-deinterleaved data to obtain output data, and thereafter, performs writing in the same order as the reading, thereby performing block deinterleaving.

Since the block deinterleaving apparatus according to the invention of Claim 6 performs the above-mentioned writing and reading operations, it enables a block deinterleaving operation on a single plane of a storage unit having a storage area of one block, and reduces the circuit scale of a storage control unit.

[0024]

According to the invention of Claim 7, the block deinterleaving apparatus defined in Claim 6 comprises a storage unit for L×M pieces of data for generating an output of the block deinterleaving apparatus; an address generation unit for generating addresses to be output to the storage unit; and a storage control unit for outputting a control signal to the storage unit; wherein, in said address generation unit, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $L^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving.

Since the block deinterleaving apparatus according to the invention of Claim 7 is constituted as described above, it optimizes the storage unit and the address generation unit, and enables block deinterleaving with a minimum circuit area.

[0025]

According to the invention of Claim 8, in the block deinterleaving apparatus defined in Claim 7, the values of α and L×M-1 are determined so that no common divisor exists between them.

Since the block deinterleaving apparatus according to the invention of Claim 8 is constituted as described above, it prevents the address generation rule from being failed, optimizes the storage unit and the address generation unit, and enables block deinterleaving with a minimum circuit area.

[0026]

According to the invention of Claim 9, in the block deinterleaving apparatus defined in Claim 7, the values of α and $L^{(-x)}$ are determined so that α is not equal to $L^{(-x)}$.

Since the block deinterleaving apparatus according to the invention of Claim 9 is constituted as described above, it prevents writing into continuous addresses at the initial writing, optimizes the storage unit and the address generation unit, and enables block deinterleaving with a minimum circuit area.

[0027]

According to the invention of Claim 10, in the block deinterleaving apparatus defined in Claim 7, the values of α, L, and M are determined at 20, 8, and 203, respectively.

Since the block deinterleaving apparatus according to the invention of Claim 10 is constituted as described above, it reduces the area of a comparator as a component of the address generation unit, optimizes the storage unit and the address generation unit, and enables block deinterleaving with a minimum circuit area.

[0028]

According to the invention of Claim 11, a block interleaving method includes performing writing while skipping addresses at the same interval corresponding to at least one address into a storage unit to which the addresses are sequentially assigned; performing reading in such order that the read data become block-interleaved data to obtain output data; and thereafter performing writing in the same order as the reading, thereby performing block interleaving.

Since the block interleaving method according to the invention of Claim 11 performs the above-mentioned writing and reading operations, it enables a block interleaving operation on a single plane of a storage unit having a storage area of one block, and reduces the circuit scale of a storage control unit.

[0029]

According to the invention of Claim 12, in the block interleaving method defined in Claim 11, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $M^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving.

Since the block interleaving method according to the invention of Claim 12 is constituted as described above, it optimizes a storage unit and an address generation unit, and enables block interleaving with a minimum circuit area.

[0030]

According to the invention of Claim 13, in the block interleaving method defined in Claim 12, the values of α and L×M-1 are determined so that no common divisor exists between them.

Since the block interleaving method according to the invention of Claim 13 is constituted as described above, it prevents the address generation rule from being failed, optimizes a storage unit and an address generation unit, and enables block interleaving with a minimum circuit area.

[0031]

According to the invention of Claim 14, in the block interleaving method defined in Claim 12, the values of α and $M^{(-x)}$ are determined so that α is not equal to $M^{(-x)}$.

Since the block interleaving method according to the invention of Claim 14 is constituted as described above, it prevents writing into continuous addresses at the initial writing, optimizes a storage unit and an address generation unit, and enables block interleaving with a minimum circuit area.

[0032]

According to the invention of Claim 15, in the block interleaving method defined in Claim 12, the values of α, L, and M are determined at 20, 8, and 203, respectively.

Since the block interleaving method according to the invention of Claim 15 is constituted as described above, it reduces the area of a comparator as a component of an address generation unit, optimizes a storage unit and the address generation unit, and enables block interleaving with a minimum circuit area.

[0033]

According to the invention of Claim 16, a block deinterleaving method includes performing writing while skipping addresses at the same interval corresponding to at least one address, into a storage unit to which the addresses are sequentially assigned; performing reading in such order that the read data become block-deinterleaved data to obtain output data; and thereafter, performing writing in the same order as the reading, thereby performing block deinterleaving.

Since the block deinterleaving method according to the invention of Claim 16 performs the above-mentioned writing and reading operations, it enables a block deinterleaving operation on a single plane of a storage unit having a storage area of one block, and reduces the circuit scale of a storage control unit.

[0034]

According to the invention of Claim 17, in the block deinterleaving method according to Claim 16, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $L^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving.

Since the block deinterleaving method according to the invention of Claim 17 is constituted as described above, it optimizes a storage unit and an address generation unit, and enables block deinterleaving with a minimum circuit area.

[0035]

According to the invention of Claim 18, in the block deinterleaving method defined in Claim 17, the values of α and L ×M-1 are determined so that no common divisor exists between them.

Since the block deinterleaving method according to the invention of Claim 18 is constituted as described above, it prevents the address generation rule from being failed, optimizes a storage unit and an address generation unit, and enables block deinterleaving with a minimum circuit area.

[0036]

According to the invention of Claim 19, in the block deinterleaving method defined in Claim 17, the values of α and $L^{(-x)}$ are determined so that α is not equal to $L^{(-x)}$.

Since the block deinterleaving method according to the invention of Claim 19 is constituted as described above, it prevents writing into continuous addresses at the initial writing, optimizes a storage unit and an address generation unit, and enables block deinterleaving with a minimum circuit area.

[0037]

According to the invention of Claim 20, in a block deinterleaving method defined in Claim 17, the values of α, L, and M are determined at 20, 8, and 203, respectively.

Since the block deinterleaving method according to the invention of Claim 20 is constituted as described above, it reduces the area of a comparator as a component of an address generation unit, optimizes a storage unit and the address generation unit, and enables block deinterleaving with a minimum circuit area.

[0038]

[Embodiments of the Invention]

(Embodiment 1)

Hereinafter, a first embodiment of the present invention will be described with reference to the drawings.

A block interleaving apparatus and a block interleaving method according to the present invention will be described.

A block interleaving apparatus and a block interleaving method according to this first embodiment aim at reducing the area or power consumption of a control unit for a storage unit, by optimizing an address generation unit included in the storage unit.

[0039]

Figure 1 is a block diagram illustrating a block interleaving apparatus which performs block interleaving of L×M pieces of data, according to the first embodiment of the invention. In figure 1, reference numeral 101 denotes an input terminal of input data to be block-interleaved by this block interleaving apparatus; 114 denotes an input terminal of a head input data sync signal which is inputted in synchronization with each block head input data of the input data to be block-interleaved; 106 denotes an input terminal of a sync signal which is generated for every input data; 112 denotes a control unit for controlling a storage unit 104 in accordance with the sync signal supplied from the sync signal input terminal 106; 103 denotes an address generation unit for generating addresses of the storage unit 104 on the basis of the sync signal supplied from the input terminal 106, and the head input data sync signal supplied from the input terminal 114; 120 denotes an output terminal from which the addresses generated by the address generation unit 103 are outputted; and 104 denotes a storage unit which performs block interleaving by writing the input data from the input terminal 101 into the addresses generated by the address generation unit 103 and reading the same, under control of the control unit 112. Reference numeral 105 denotes an output terminal for outputting the data interleaved by this block interleaving apparatus.

[0040]

Further, in the address generation unit 103 shown in figure 1, reference numeral 110 denotes a constant generator for generating a constant M; 111 denotes a multiplier for multiplying the output of the register 113 by the initial value M; 140 denotes an overflow processing unit to be used when the output from the multiplier 111 overflows; 113 denotes a register in which the output of the overflow processing unit 140 or the output of the constant generator 118 is set; 118 denotes a constant generator for generating an initial value α; 115 denotes an adder for adding the output of the register 113 and the output of the register 117; 141 denotes a overflow processing unit to be used when the output from the multiplier 111 overflows; 117 denotes a register in which the output of the overflow processing unit 141 or the output of a constant generator 119 is set; and 119 denotes a constant generator for generating an initial value 1.

Figure 2 is a diagram schematically illustrating the operation of the block interleaving apparatus according to the first embodiment, wherein 4 rows × 5 columns of data are subjected to block interleaving.

[0041]

Next, the operation shown in figure 1 will be described taking, as an example, the case where 4 rows × 5 columns of data shown in figure 2 are subjected to block interleaving.

As shown in figure 1, the block interleaving apparatus of this first embodiment performs block interleaving by writing the data supplied from the input terminal 101 into the L×M data storage unit 104, and reading the data from the L×M data storage unit 104. At this time, the control unit 112 outputs a control signal to the storage unit 104 to control the writing and reading so that the writing and reading are carried out in the order shown in figure 2, and the address generation unit 103 generate addresses for the writing and reading and outputs the addresses to the storage unit 104, whereby a block-interleaved output 105 can be generated by using a single plane of the storage unit having a storage area corresponding to one block.

[0042]

Assuming that the addresses of the storage unit 104 of the block interleaving apparatus are allocated as shown in figure 13(a), initially, REG is set at 2 as shown in figure 2(a), and a writing address which increases by 2 for every input data is sequentially generated, with address 0 shown in figure 13(a) being an initial value. At this time, when the writing address exceeds 19 (=4×5-1), this 19 is subtracted. Then, according to the writing addresses which are generated under this address generation rule, data writing is performed until accesses to all the addresses in the block are completed.

To be specific, while in the conventional method shown in figure 13(a) data are written in the order of 0 → 1 → 2 → ... → 19, i.e., in the ascending order of the addresses, in this first embodiment data are written in every other address.

[0043]

Next, as shown in figure 2(b), the REG is multiplied by 5, and an address which increases by 10 (=2×5) for every input data is sequentially generated, with the address allocation shown in figure 2(a) as a reference, and address 0 in figure 2(a) as an initial value. At this time, when the address exceeds 19 (=4×5-

1), this 19 is subtracted.

Then, reading and writing are performed according to the addresses generated under the address generation rule until accesses to all the addresses in the block are completed.

[0044]

Next, as shown in figure 2(c), the REG is multiplied by 5. Since the value exceeds 19, division is carried out using this 19 to obtain a remainder 12.

Then, an address which increases by 12 for every input data is sequentially generated with the address arrangement shown in figure 13(a) as a reference, and address 0 as an initial value. When the address exceeds 19 (=4×5-1), this 19 is subtracted.

Then, reading and writing are carried out according to the addresses generated under the address generation rule, until accesses to all the addresses in the block are completed.

[0045]

Thereafter, reading and writing are sequentially performed in different address orders, whereby, in this example, the address order returns to that shown in figure 2(a) at the point of time shown in figure 2(j).

By repeating the above-described procedure, block interleaving can be carried out using only a single plane of the storage unit having a storage area of 1 block, as shown in figure 3. This block interleaving is realized by contriving the writing and reading control by the control unit 112 and the addresses of the storage unit 104 generated by the address generation unit 103 as described above. In addition, in this first embodiment, the circuit scale and power consumption of the address generation unit can be reduced.

[0046]

The address generation rule according to the first embodiment is as follows.

Assuming that the n-th address is Ab(n), the number of rows of the storage unit is L, the number of columns is M, the block number b is an integer not less than 0, and x is an arbitrary integer not less than 0 and not larger than b,

$$Ab(n)=(Ab(n-1)+\alpha \times M^{**}(b-x))\bmod(L \times M-1) \quad ...(3)$$

Further,

$$REG=\alpha \times M^{**}(b-x)\bmod(L \times M-1)$$

wherein Ab(0) is 0, α is an integer not less than 2, and ** indicates a power.

[0047]

Accordingly, in the above example, the first writing is performed on every other address by setting $\alpha = 2$. Although data writing in every third or more address is also possible by appropriately setting the value of α, a common divisor should not exist between α and L×M-1. The reason is as follows. When a common divisor exists between α and L×M-1, even though the last data amongst the data within the block should be written in address L×M-1, an address becomes L×M-1 in the middle of the processing, whereby the address generation rule fails.

Further, α should not be equal to the (-X)th power of M. This case corresponds to the conventional example and, therefore, further reductions in circuit scale and power consumption cannot be achieved.

[0048]

Next, a description will be given of the address generating operation of the address generation unit, which is required for the above-described writing and reading.

In the address generation unit 103 shown in figure 1, the constant generator 118 generates an initial value "α" and outputs it to the register 113. The multiplier 111 multiplies the output of the register 113 by the output "M" from the constant generator 110 and outputs the result to the overflow processing unit 140.

[0049]

The overflow processing unit 140 repeats subtraction of "L×M-1" until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 113. The register 113 is updated to the output value of the overflow processing unit 140 by the block head input data sync signal 114 only after the L×M pieces of data are inputted.

[0050]

Further, the constant generator 119 generates an initial value "1" and outputs it to the register 117. The adder 115 adds the output of the register 117 and the output of the register 113, and outputs the result to the overflow processing unit 141.

[0051]

The overflow processing unit 141 repeats subtraction of "L×M-1" until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 117. The register 117 is reset to the initial value "1" by the block head input data sync signal 114 when L×M pieces of data have been input, and it is updated for every input data by the sync signal 106.

[0052]

Thereby, the address generation unit generates, with the 0th address Ab(0) of a block having a block number b being set at 0, the n-th (n: integer, 0≤n) address Ab(n) of this block b from the remainder which is left when dividing the sum of the product of α (α: integer, 2≤α) and $M^{(b-x)}$ (x: integer, 0≤x≤b) and Ab(n-1) by L×M-1, thereby generating addresses of the storage unit according to the first embodiment, and the overflow processing unit prevents the numerical values in the address generation unit from diverging over L×M-1 in the address generation unit due to repetition of multiplication and addition.

[0053]

Figure 4 shows the constructions of the overflow processing units 140 and 141. In the overflow processing unit 140 shown in figure 4(a), reference numeral 121 denotes a switch which selects and outputs either the input to the overflow processing unit or the output of the register 125 according to the block head input data sync signal 114. Reference numeral 123 denotes a comparator for judging whether the input is equal to or larger than L×M-1 or not, 122 denotes a subtracter for subtracting L×M-1 from the input, and 124 denotes a switch which selects and outputs either the output of the switch 121 or the output of the subtracter 122 according to the result of judgement by the comparator 123. Reference numeral 125 denotes a register which receives the output of the switch 124.

[0054]

Further, in the overflow processing unit 141 shown in figure 4(b), reference numeral 131 denotes a switch which selects and outputs either the input to the overflow processing unit or the output of the register 135 according to the block head input data sync signal 114. Reference numeral 133 denotes a comparator for judging whether the input is equal to or larger than L×M-1 or not, 132 denotes a subtracter for subtracting L×M-1 from the input, and 134 denotes a switch which selects and outputs either the output of the switch 131 or the output of the subtracter 132 according to the result of judgement by the comparator 133. Reference numeral 135 denotes a register which receives the output of the switch 134.

[0055]

Next, the operation will be described. Since the operations of the overflow processing units 140 and 141 are identical to each other, only the operation of the overflow processing unit 140 will be described. Initially, the switch 121 is given the input of the overflow processing unit 140 and the output of the selector 125. When the input data corresponds to the head of the block and a block head input data sync signal 114 is input, the switch 121 selects the input of the overflow processing unit 140. In other cases, it selects the output of the register 125. The output of the switch 121 is compared with L×M-1 by the comparator 123. The switch 124 is given the output of the subtracter 122 which subtracts L×M-1 from the output of the switch 121, and the output of the switch 121. When the comparator 123 decides that the output of the switch 121 is equal to or larger than L×M-1, the switch 124 selects the output of the subtracter 122. In other cases, the switch 124 selects the output of the switch 121, and updated for every input data by the input data sync signal 106. Thereby, when the input to the overflow processing unit 140 is equal to or larger than L×M-1, the overflow processing unit 140 repeats subtraction of L×M-1 so that the input is kept smaller than L×M-1.

The overflow processing unit prevents the numerical values from diverging over L×M-1 due to repetition of multiplication and addition in the address generation unit.

[0056]

Further, the initial state can be changed to an arbitrary state other than that shown in figure 2(a) by setting a value of x in formula (3). Also in this case, the state of the block returns to the initial state by repeating the above-described processing and, thereafter, the same repetition takes place.

[0057]

As described above, the first embodiment is able to perform block interleaving by using a storage unit having a storage area corresponding to one block, like the prior art. However, this first embodiment is further able to reduce the circuit scale of the address generation unit.

[0058]

Hereinafter, this advantage will be described.

Table 1 shows the transition of the value of the register 113 when the prior art apparatus is constituted with the same circuit structure as that of the first embodiment (in figure 1, the value of α of the constant generator 118 is set at "1" in the prior art while it is set at "2" or more in the first embodiment).

[0059]

[Table 1]

```

------------------
L=      4
M=      5
 α=     1
------------------

val=      1 →      5
val=      5 →     25     6
val=      6 →     30    11
val=     11 →     55    36     17
val=     17 →     85    66     47     28     9
val=      9 →     45    26      7
val=      7 →     35    16
val=     16 →     80    61     42     23     4
val=      4 →     20     1

overtime    =     16
maxoverval  =     85
minoverval  =     20
maxval      =     17

------------------
L=      4
M=      5
 α=     2
------------------

val=      2 →     10
val=     10 →     50    31     12
val=     12 →     60    41     22      3
val=      3 →     15
val=     15 →     75    56     37     18
val=     18 →     90    71     52     33    14
val=     14 →     70    51     32     13
val=     13 →     65    46     27      8
val=      8 →     40    21      2

overtime    =     20
maxoverval  =     90
minoverval  =     21
maxval      =     18
```

[0060]

Table 1 shows the transition of the value of the register 113 when L=4 and M=5, i.e., block interleaving is performed on 4 rows × 5 columns of data. In table 1, "val" indicates the value of the register 113, and when the val exceeds the threshold value

"19" (= 5×4-1), this value is gradually decreased by the overflow processing unit such that it falls within the threshold value.

[0061]

Further, "overtime" indicates the number of times the value of the register 113 exceeds the threshold value, "maxoverval" indicates the maximum value among the values of the register 113 which exceed the threshold value, "minoverval" is the minimum value among the values of the register 113 which exceed the threshold value, and "maxval" indicates the maximum value of the register 113.

[0062]

Further, the 8th to 16th rows on table 1 indicate the transition of the value of the register 113 according to the prior art (α = 1 in the 5th row), and the 29th to 37th rows indicate the transition of the value of the register 113 according to the first embodiment (α = 2 in the 26th row).

[0063]

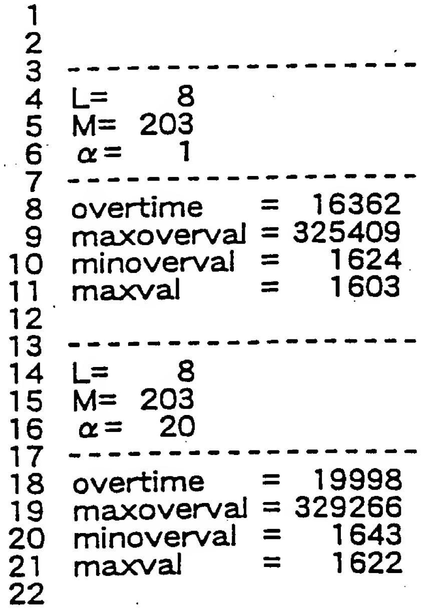
When these transitions are compared with each other, it is found that, in the prior art, the minimum value "minoverval" among the values of the register 113 which exceed the threshold value is 20 (= the value of L×M, i.e., the minimum value which exceeds the threshold value "19") while it is 21 in this first embodiment, i.e., larger than that of the prior art.

[0064]

The 3rd to 21st rows on table 2 show the calculation results of the values of the register 113 in the case where L=8 and M=23, that is, block interleaving is performed on 8 rows × 203 columns of data. The 8th to 11th rows on table 2 show the calculation results of the values of the register 113 according to the prior art, while the 18th to 21st rows on table 2 show the calculation results of the values of the register 113 according to the first embodiment.

[0065]

[Table 2]



```
 1
 2
 3 -------------------
 4 L=     8
 5 M=   203
 6  α=     1
 7 -------------------
 8 overtime     =  16362
 9 maxoverval = 325409
10 minoverval =    1624
11 maxval       =    1603
12
13 -------------------
14 L=     8
15 M=   203
16  α=    20
17 -------------------
18 overtime     =  19998
19 maxoverval = 329266
20 minoverval =    1643
21 maxval       =    1622
22
```

[0066]

When these calculation results are compared with each other, it is found that, in the prior art, the minimum value "minoverval" among the values of the register 113 which exceed the threshold value of the overflow processing unit 140 is "1624" (= the value of L×M, i.e., the minimum value which exceeds the threshold value "1623") while it is "1643" in the first embodiment, that is, larger than that of the prior art.

[0067]

As described above, according to the first embodiment, when performing writing or reading in/from the storage unit, the first writing is performed on every second (or more) address, while it is performed successively on every address in the prior art. Since the address order in the first writing is different from that of the prior art, the minimum value which exceeds the threshold value and is stored in the register 113 becomes equal to or larger than that of the prior art.

[0068]

Therefore, while a comparator for comparing the values larger than "1624" is needed in the prior art, a comparator for comparing the values larger than "1643" is needed in this first embodiment, and therefore, the function of the comparator is simplified in this first embodiment.

As described above, when the threshold value to be compared with the input by the comparator in the overflow processing unit can be made larger than L×M, the circuit scale of the comparator can be surely reduced as compared with that of the prior art.

[0069]

Hereinafter, this advantage will be described taking, as an example, an apparatus which performs block interleaving on 8 rows × 203 columns of data.

In this case, according to the prior art method, the comparator 123 in the overflow processing unit 140 must detect that the input is equal to or larger than L×M, i.e., "1624".

Figure 5 shows the structure of the comparator in the overflow processing unit of the apparatus which performs block interleaving on 8 rows × 203 columns of data by the prior art method.

In figure 5, 3311～3319 and 3321～3333 denote AND gates, and 3336～3339 and 3350～3356 denote OR gates.

[0070]

Next, the operation will be described. In order to decide that the input I is equal to or larger than "1624", it is necessary to judge that the bit pattern of the input I is equal to or larger than "011001011000" which is obtained by expanding 1624 into binary digits. At this time, whether the lower three bits of the input I are "0" or "1" does not influence the decision, and when all of the lower three bits are "1", the input value is "1631". Accordingly, the lower three bits are not inputted when deciding that the input value is "1624", whereby it is possible to judge that the input value is "1624"～"1631".

[0071]

The AND gates 3311～3319 decide that the input value is "1624"～"1631" according to the above-mentioned principle, and the AND gate 3311～3314 output "1" when the bit pattern from the 12th bit to the 5th bit of the input value matches "01100101100". The AND gates 3315～3316 output "1" when all of the outputs from the AND gates 3311～3314 are "1", and the AND gates 3317 outputs "1" when all of the outputs from the AND gates 3315 and 3316 are "1". Further, the AND gate 3318 outputs "1" when the 4th bit of the input value is "1" and the output of the AND gate 3316 is "1". Further, the AND gate 3319 outputs "1" when all of the outputs from the AND gates 3317 and 3318 are "1". Accordingly, when the output from the AND gate 3319 is "1", it becomes clear that the input value is "1624"～"1631".

[0072]

Likewise, the AND gates 3321～3326 decide that the input is "1632"～"1663". The AND gates 3327～3330 decide that the input is "1664"～"1791". The AND gates 3331～3333 decide that the input is "1792"～"2047". Further, the OR gates 3350～3356 decide that the input is "2048"～"524287" (values up to "524287" are decided because maxoverval is "325409").

Accordingly, by integrating these results of decisions by the OR gates 3336～3339, the comparator can decide that the input value is equal to or larger than "1624".

[0073]

As described above, while the comparator of the prior art apparatus must decide that the input is equal to or larger than L ×M, i.e., "1624", the comparator of this first embodiment decides that the input is equal to or larger than "1643", as can be seen in comparison between the 1st to 11th rows on table 2 and the 13th to 21st rows on table 2.

[0074]

Figure 6 shows the structure of the comparator in the overflow processing unit of the block interleaving apparatus according to the first embodiment of the invention.

In figure 6, 3321～3333 denote AND gates, and 3340～3342 and 3350～3356 denote OR gates.

In figure 6, the comparator should decide that the input is equal to or larger than "1643". However, since this decision is included in the decision of "1632" and larger values, this circuit decides that the input is equal to or larger than "1632".

[0075]

Initially, the AND gates 3321～3326 decide that the input is "1632"～"1663". The AND gates 3327～3330 decide that the input is "1664"～"1791". The AND gates 3331～3333 decide that the input is "1792"～"2047". Further, the OR gates 3350～3356 decide that the input is "2048"～"524287" (values up to "524287" are decided because maxoverval is "329266").

Accordingly, by integrating these results of decisions by the OR gates 3340～3342, the comparator can decide that the input value is equal to or larger than "1632", i.e., "1643".

[0076]

The circuit shown in figure 6 requires thirteen AND gates and ten OR gates while the prior art circuit shown in figure 5 requires twenty-two AND gates and eleven OR gates. That is, the circuit shown in figure 6 is reduced in circuit scale as compared with the prior art circuit because the objects to be compared are reduced, resulting in reduced area and reduced power consumption.

[0077]

By the way, the block interleaving with L=8, M=203, and α=20 is effectively used for error correction in BS digital broadcasting.

To be specific, in the case of BS digital broadcasting, one data segment to be a target of correction by a Reed-Solomon decoder has 203 bytes in a data interleaving apparatus and, if the number of columns in a block interleaving apparatus at the transmitting end is 203, the correction ability of the Reed-Solomon decoder can be improved with the least storage capacity of the interleaving apparatus. Further, as the numbers of rows and columns are increased, the correction ability of the Reed-Solomon decoder against continuous burst errors is improved.

[0078]

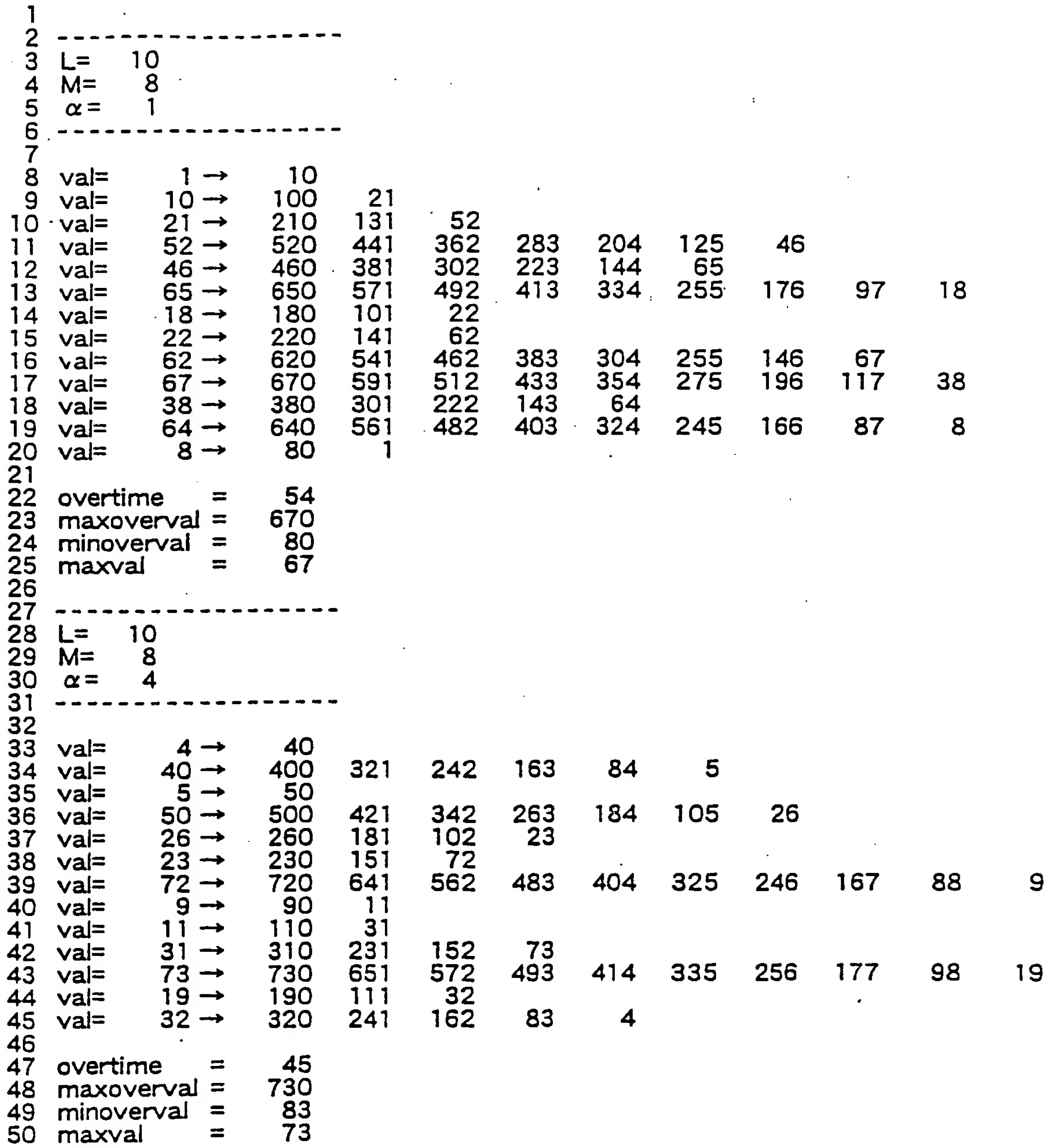
Further, α may be an arbitrary integer not less than 2 so long as there is no common divisor between α and L×M-1 and α is not equal to $M^{(-x)}$, and the greatest effect is obtained when α is 20.

Further, there is a case where the power consumption can be reduced according to a principle different from that described above.

Hereinafter, this case will be described. Table 3 shows the transition of the values of the register 113 when performing block interleaving with L=10 and M=8, i.e., on 10 rows × 8 columns of data, wherein the transition according to the first embodiment is contrasted with that according to the prior art.

[0079]

[Table 3]



```

------------------
L=    10
M=     8
 α=    1
------------------

val=      1 →    10
val=     10 →   100   21
val=     21 →   210  131   52
val=     52 →   520  441  362  283  204  125   46
val=     46 →   460  381  302  223  144   65
val=     65 →   650  571  492  413  334  255  176   97   18
val=     18 →   180  101   22
val=     22 →   220  141   62
val=     62 →   620  541  462  383  304  255  146   67
val=     67 →   670  591  512  433  354  275  196  117   38
val=     38 →   380  301  222  143   64
val=     64 →   640  561  482  403  324  245  166   87    8
val=      8 →    80    1

overtime     =   54
maxoverval =    670
minoverval  =    80
maxval       =   67

------------------
L=    10
M=     8
 α=    4
------------------

val=      4 →    40
val=     40 →   400  321  242  163   84    5
val=      5 →    50
val=     50 →   500  421  342  263  184  105   26
val=     26 →   260  181  102   23
val=     23 →   230  151   72
val=     72 →   720  641  562  483  404  325  246  167   88    9
val=      9 →    90   11
val=     11 →   110   31
val=     31 →   310  231  152   73
val=     73 →   730  651  572  493  414  335  256  177   98   19
val=     19 →   190  111   32
val=     32 →   320  241  162   83    4

overtime     =   45
maxoverval =    730
minoverval  =    83
maxval       =   73
```

[0080]

As is evident from table 3, while in the prior art the number of times the value of the register 113 exceeds the threshold value (overtime) is 54, in this first embodiment it is reduced to 45. This reduction in the overtime results in a reduction in the computational complexity of the overflow processing unit 140. Further, while in the prior art the number of overflow times of the overflow processing unit 141 is 474, in this first embodiment it is reduced to 395, resulting in a reduction in the computational complexity of the overflow processing unit 141.

Thereby, reduction in power consumption is realized.

[0081]

As described above, the block interleaving apparatus according to the first embodiment of the invention is provided with the L×M data storage unit which generates an output from the block interleaving apparatus, the address generation unit which outputs addresses to the storage unit, and the storage unit control unit which outputs a control signal to the storage unit. In the address generation unit, the 0th address Ab(0) of a block having a block number b is set at 0, and the n-th (n: integer, $0 \leq n$) address Ab(n) of this block is generated from the remainder which is left when dividing the sum of the product of α (α: integer, $2 \leq \alpha$) and $M^{(b-x)}$ (x: integer, $0 \leq x \leq b$) and Ab(n-1) by L×M-1, and reading and writing are repeated from/in the generated address, thereby performing block interleaving. Therefore, the storage unit and the address generation unit can be optimized, and the block interleaving can be performed with the minimum circuit scale.

[0082]

Further, since the first address and the last address of each block are constant, two pieces of data in these addresses can be processed simultaneously by allocating a continuous area of the storage unit to these addresses, whereby the number of accesses to the storage unit is reduced, resulting in reduced power consumption of the address generation unit.

[0083]

Further, especially when performing block interleaving with L=8 and M=203, in the prior art address generation unit disclosed in Japanese Published Patent Application No. Hei.8-511393, the 0th address Ab(0) of a block having a block number b is set at 0, and the n-th (n: integer, $0 \leq n$) address Ab(n) of this block is generated from the remainder which is left when dividing the sum of the product of $M^{(b-x)}$ (x: integer, $0 \leq x \leq b$) and Ab(n-1) by L×M-1. In repetition of this calculation, the target value to be divided increases infinitely. So, when implementing this calculation by a circuit, the circuit is composed of a multiplier which sets the initial value at $M^{(b-x-1)}$, multiplies the input by M, and outputs the product to an overflow processing unit 1 (hereinafter referred to as a remainder generator 1); the remainder generator

1 which outputs the remainder obtained by dividing the input by L×M-1, to the multiplier and an adder; the adder which adds Ab(n-1) to the output from the remainder generator 1 and outputs the sum to an overflow processing unit 2 (hereinafter, referred to as a remainder generator 2); and the remainder generator 2 which generates Ab(n) as the remainder obtained by dividing the input by L×M-1. The remainder generator 1 is composed of a comparator and a subtracter for subtracting L×M-1 from the input until the input becomes equal to or lower than L×M-1. In this case, since the minimum value to be subjected to the subtraction is "1624", the comparator should be provided with the function of deciding "1624" and larger values.

[0084]

However, in the block interleaving apparatus of this first embodiment, assuming that α=20, L=8, and M=203, the circuit is composed of a multiplier which sets the initial value at $M^{(b-x-1)} \times \alpha$, multiplies the input by M, and outputs the product to a remainder generator 1; the remainder generator 1 which outputs the remainder obtained by dividing the input by L×M-1, to the multiplier and an adder; the adder which adds Ab(n-1) and the output from the remainder generator 1 and outputs the sum to a remainder generator 2; and the remainder generator 2 which generates Ab(n) as the remainder obtained by dividing the input by L×M-1. The remainder generator 1 is composed of a comparator and a subtracter for subtracting L×M-1 from the input until the input becomes equal to or lower than L×M-1. In this case, since the minimum value to be subjected to the subtraction is "1643", the comparator is required of the function of deciding "1643" and larger values, whereby the area of the comparator is reduced, and the block interleaving can be performed with the minimum circuit area.

[0085]

Further, it is also possible to realize block interleaving by setting the reading address and the writing address at Ab(n) and Ab(n-t) (t: natural number, $t \leq L \times M-2$), respectively, and repeating reading and writing from/in each address at each point of time.

Furthermore, when Ab(0) is set at β (β: natural number, $\beta \leq L \times M-1$), an address Ab(n) may be generated from the remainder which is left when dividing the sum of the product of α and $M^{(b-x)}$ and Ab(n-1) by L×M-1.

[0086]

(Embodiment 2)

Hereinafter, a second embodiment of the present invention will be described with reference to the drawings.

Initially, a block deinterleaving apparatus and a block deinterleaving method according to the present invention will be described.

In a block deinterleaving apparatus and a block deinterleaving method according to this second embodiment, an address generation unit included in a storage unit is optimized to reduce the area or power consumption of a control unit for the storage unit.

[0087]

Figure 7 is a block diagram illustrating a block deinterleaving apparatus which performs block deinterleaving of L ×M pieces of data, according to the second embodiment of the invention. In figure 7, reference numeral 1 denotes an input terminal of input data to be block-deinterleaved by this block deinterleaving apparatus; 14 denotes an input terminal of a head input data sync signal which is inputted in synchronization with block head input data of the input data to be block-deinterleaved; 12 denotes a control unit for controlling a storage unit 4 in accordance with the sync signal supplied from the sync signal input terminal 6; 3 denotes an address generation unit for generating addresses of the storage unit 4 on the basis of the sync signal supplied from the input terminal 6 and the head input data sync signal supplied from the input signal 14; 20 denotes an output terminal from which the addresses generated by the address generation unit 3 are outputted; 4 denotes a storage unit for performing block deinterleaving by writing the input data supplied from the input terminal 1 into the addresses generated by the address generation unit 3 and reading the data, under control of the control unit 12. Reference numeral 5 denotes an output terminal for outputting the data deinterleaved by this block deinterleaving apparatus.

[0088]

In the address generation unit 3 shown in figure 7, reference numeral 10 denotes a constant generator for generating a constant L; 11 denotes a multiplier for multiplying an output signal from a register 13 by the constant L; 40 denotes an overflow processing unit for performing processing when the output from the multiplier 11 overflows; 13 denotes a register in which the output of the overflow processing unit 40 or the output of the constant generator 18 is set; 18 denotes a constant generator for generating an initial value α; 15 denotes an adder for adding the output signal of the register 13 and the output signal of the register 17; 41 denotes an overflow processing unit for performing processing when the output from the adder 15 overflows; 17 denotes a register in which the output of the overflow processing unit 41 or the output of the constant generator 19 is set; and 19 denotes a constant generator for generating an initial value "1".

Figure 8 is a diagram schematically illustrating the operation of the block deinterleaving apparatus according to the second embodiment, taking, as an example, a case where 4 rows × 5 columns of data are subjected to block deinterleaving.

[0089]

Next, the operation of figure 7 will be described taking, as an example, the case where 4 rows × 5 columns of data shown in figure 8 are subjected to block deinterleaving.

As shown in figure 8, the block deinterleaving apparatus according to the second embodiment writes the input data supplied from the input terminal 1 in the L×M data storage unit 4, and reads the data from the L×M data storage unit 4, thereby performing block deinterleaving. At this time, in order to perform the writing and reading in the order as shown in figure 8, the control unit 12 controls the writing and reading by outputting a control signal to the storage unit 4, and the address generation unit 3 generates addresses for the writing and reading and outputs the addresses to the storage unit 4, thereby generating an output 5 which is block-deinterleaved by a single plane of a storage unit having a storage area corresponding to one block.

[0090]

Assuming that addresses of the storage unit 4 of the block deinterleaving apparatus are allocated as shown in figure 13(a), initially, REG is set at 2 as shown in figure 8(a), and a writing address which increases by 2 for each input data with address 0 shown in figure 13(a) as an initial value, is successively generated. At this time, when the writing address exceeds 19 (=4 ×5-1), this 19 is subtracted. Then, according to the writing addresses generated on the basis of the address generation rule, data writing is performed until accesses to all the addresses in the block are completed.

While in the prior art method shown in figure 13(a) data are successively written in the order of 0→1→2→...→19, i.e., according to the address increment, in this second embodiment data are written in every other address.

[0091]

Next, as shown in figure 8(b), the REG is multiplied by 4, and an address which increases by 8 (=2×4) for each input data is successively generated with the address arrangement shown in figure 13(a) as a reference and address 0 shown in figure 13(a) as an initial value. At this time, when the address exceeds 19 (=4×5-1), this 19 is subtracted.

Then, reading and writing are performed according to the addresses generated on the basis of the address generation rule until accesses to all the addresses in the block are completed.

[0092]

Next, as shown in figure 8(c), the REG is multiplied by 4. Since the result exceeds 19, division is performed with this 19 to obtain a remainder 13.

Then, an address which increases by 13 for each input data is successively generated with the address arrangement shown in figure 13(a) as a reference and address 0 as an initial value. When the address exceeds 19 (=4×5-1), this 19 is subtracted.

Then, reading and writing are carried out according to the addresses generated on the basis of the address generation rule until accesses to all the addresses in the block are completed.

[0093]

Thereafter, reading and writing are sequentially performed in different address orders, whereby, in this example, the address order returns to that shown in figure 8(a) at the point of time shown in figure 8(j).

By repeating the above-described procedure, block interleaving can be carried out using only a single plane of the storage unit having a storage area of one block, as shown in figure 9. This block interleaving is realized by contriving the writing and reading control by the control unit 12 and the addresses of the storage unit 4 generated by the address generation unit 3 as described above. In addition, in this second embodiment, the circuit scale and power consumption of the address generation unit can be reduced.

[0094]

The address generation rule according to the second embodiment is as follows.

Assuming that the n-th address is Ab(n), the number of rows of the storage unit is L, the number of columns is M, the block number b is an integer not less than 0 ($0 \leq b$), and x is an arbitrary integer not less than 0 and not larger than b ($0 \leq x \leq b$),

$$Ab(n)=(Ab(n-1)+\alpha \times L^{**}(b-x))\bmod(L \times M-1) \qquad ...(4)$$

Further,

$$REG=\alpha \times L^{**}(b-x)\bmod(L \times M-1)$$

wherein Ab(0) is 0, α is an integer not less than 2 ($2 \leq \alpha$), and ** indicates a power.

[0095]

Accordingly, in the above example, the first writing is performed on every other address by setting the value of α at 2. Although data writing in every third or more address is also possible by appropriately setting the value of α, a common divisor should not exist between α and $L \times M-1$. The reason is as follows. When a common divisor exists between α and $L \times M-1$, although the last data amongst the data within the block should be always written in address $L \times M-1$, the address becomes $L \times M-1$ in the middle of the processing, and the address generation rule fails.

Further, α should not be equal to $M^{(-x)}$. This case corresponds to the prior art and, therefore, further reductions in circuit scale and power consumption cannot be expected.

[0096]

Hereinafter, a description will be given of address generating operation of the address generation unit 3, required for performing the above-described writing and reading.

In the address generation unit 3 shown in figure 7, the constant generator 18 generates an initial value "α" and outputs this to the register 13. The multiplier 11 multiplies the output of the register 13 by the output "L" of the constant generator 10 and outputs the result to the overflow processing unit 40.

[0097]

The overflow processing unit 40 subtracts "L×M-1" until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 13. The register 13 is updated to the output value of the overflow processing unit 140 by the block head input data sync signal 14 only after the L×M pieces of data are inputted.

[0098]

Further, the constant generator 19 generates an initial value "1" and outputs it to the register 17. The adder 15 adds the output of the register 17 and the output of the register 13, and outputs the sum to the overflow processing unit 41.

The overflow processing unit 41 subtracts "L×M-1" until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 17. The register 17 is reset to the initial value "1" by the block head input data sync signal when L ×M pieces of data have been input, and it is updated for every input data by the sync signal input 6.

[0099]

Thereby, the address generation unit generates addresses of the storage unit by setting the 0th address Ab(0) of a block having block number b at 0, and generating the n-th (n: integer, $0 \leq n$) address Ab(n) of this block from the remainder which is left when dividing the sum of the product of α (α: integer, $2 \leq \alpha$) and $M^{(b-x)}$ (x: integer, $0 \leq x \leq b$) and Ab(n-1) by L×M-1, and the overflow processing unit prevents the numerical value in the address generation unit from diverging over L×M-1 due to repetition of multiplication and addition in the address generation unit.

[0100]

Figure 10 shows the constructions of the overflow processing units 40 and 41. In the overflow processing unit 40 shown in figure 10(a), reference numeral 21 denotes a switch which selects and outputs either the input to the overflow processing unit or the output of the register 25 according to the block head input data sync signal 14. Reference numeral 23 denotes a comparator for judging whether the input is equal to or larger than L×M-1 or not, 22 denotes a subtracter for subtracting L×M-1 from the input, and 24 denotes a switch which selects and outputs either the output of the switch 21 or the output of the subtracter 22 according to the result of judgement by the comparator 23. Reference numeral 25 denotes a register which receives the output of the switch 24.

[0101]

Further, in the overflow processing unit 41 shown in figure 10(b), reference numeral 31 denotes a switch which selects and outputs either the input to the overflow processing unit or the output of the register 35 according to the block head input data sync signal 14. Reference numeral 33 denotes a comparator for judging whether the input is equal to or larger than L×M-1 or not, 32 denotes a subtracter for subtracting L×M-1 from the input, and 34 denotes a switch which selects and outputs either the output of the switch 31 or the output of the subtracter 32 according to the result of judgement by the comparator 33. Reference numeral 35 denotes a register which receives the output of the switch 34.

[0102]

Next, the operation will be described. Since the operations of the overflow processing units 40 and 41 are identical to each other, only the operation of the overflow processing unit 40 will be described. Initially, the switch 21 is given the input of the overflow processing unit 40 and the output of the selector 25. When the input data corresponds to the head of the block and a block head input data sync signal 14 is input, the switch 21 selects the input of the overflow processing unit 40. In other cases, it selects the output of the register 25. The output of the switch 21 is compared with L×M-1 by the comparator 23. The switch 24 is given the output of the subtracter 22 which subtracts L×M-1 from the output of the switch 21, and the output of the switch 21. When the comparator 23 decides that the output of the switch 21 is equal to or larger than L×M-1, the switch 24 selects the output of the subtracter 22. In other cases, the switch 24 selects the output of the switch 21. The output of the switch 24 is output to the register 25, and updated for every input data by the input data sync signal 6. Thereby, when the input to the overflow processing unit 40 is equal to or larger than L×M-1, the overflow processing unit 40 repeats subtraction of L×M-1 so that the input is kept smaller than L×M-1.

The overflow processing unit prevents the numerical values from diverging over L×M-1 due to repetition of multiplication and addition in the address generation unit.

[0103]

Further, the initial state can be changed to an arbitrary state other than that shown in figure 8(a) by setting a value of x in formula (4). Also in this case, the state of the block returns to the initial state by repeating the above-described processing and, thereafter, the same repetition takes place.

As described above, the second embodiment is able to perform block interleaving by using a storage unit having a storage area corresponding to one block, like the prior art. However, this second embodiment is further able to reduce the circuit scale of the address generation unit.

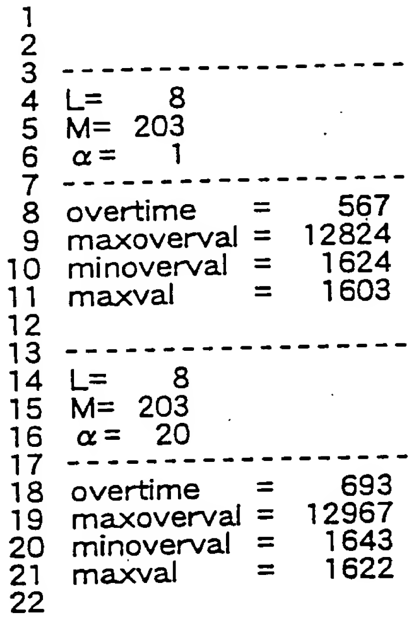
[0104]

Hereinafter, this advantage will be described with reference to table 4.

The 1st to 21st rows on table 4 show the calculation results of the values of the register 13 in the case where L(=8)×M(=203) pieces of data (i.e., 8 rows × 203 columns of data) are subjected to block deinterleaving. The 8th to 11th rows on table 4 shows the calculation results of the values of the register 13 according to the prior art, and the 18th to 21st rows on table 4 show the calculation results of the values of the register 13 according to the second embodiment.

[0105]

[Table 4]



```
------------------
L=    8
M=  203
 α=    1
------------------
overtime     =    567
maxoverval =  12824
minoverval  =   1624
maxval        =   1603

------------------
L=    8
M=  203
 α=   20
------------------
overtime     =    693
maxoverval =  12967
minoverval  =   1643
maxval        =   1622
```

[0106]

When these calculation results are compared with each other, it is found that, in the prior art, the minimum value (minoverval) among the values of the register 13 which exceed the threshold value of the overflow processing unit 40 is "1624" (= the value of L×M, i.e., the minimum value which exceeds the threshold value "1623"), while it is "1643" in this second embodiment, that is, larger than that of the prior art.

[0107]

As described above, according to the second embodiment, when performing writing or reading in/from the storage unit, the first writing is performed on every second (or more) address, while it is performed successively on every address in the prior art.

Since the address order in the first writing is different from that of the prior art, the minimum value which exceeds the threshold value and is stored in the register 13 becomes equal to or larger than that of the prior art.

[0108]

Therefore, while a comparator for comparing the values larger than "1624" is needed in the prior art, a comparator for comparing the values larger than "1643" is needed in this first embodiment, and therefore, the function of the comparator is simplified in this first embodiment.

As described above, when the threshold value to be compared with the input by the comparator in the overflow processing unit can be made larger than L×M, the circuit scale of the comparator can be surely reduced as compared with that of the prior art.

[0109]

Hereinafter, this advantage will be described taking, as an example, an apparatus which performs block interleaving on 8 rows × 203 columns of data.

In this case, according to the prior art method, the comparator 23 in the overflow processing unit 40 must detect that the input is equal to or larger than L×M, i.e., "1624".

[0110]

Figure 11 shows the structure of the comparator in the overflow processing unit of the apparatus which performs block interleaving on 8 rows × 203 columns of data by the prior art method.

In figure 11, 2311～2319 and 2321～2333 denote AND gates, and 2334～2339 denote OR gates.

[0111]

Next, the operation will be described. In order to decide that the input I is equal to or larger than "1624", it is necessary to judge that the bit pattern of the input I is equal to or larger than "011001011000" which is obtained by expanding 1624 into binary digits. At this time, whether the lower three bits of the input I are "0" or "1" does not influence the decision, and when all of the lower three bits are "1", the input value is "1631". Accordingly, the lower three bits are not inputted when deciding that the input value is "1624", whereby it is possible to judge that the input value is "1624"～"1631".

[0112]

The AND gates 2311～2319 decide that the input value is "1624"～"1631" according to the above-mentioned principle, and the AND gate 2311～2314 output "1" when the bit pattern from the 12th bit to the 5th bit of the input value matches "01100101100". The AND gates 2315～2316 output "1" when all of the outputs from the AND gates 2311～2314 are "1", and the AND gates 2317 outputs "1" when all of the outputs from the AND gates 2315 and 2316 are "1". Further, the AND gate 2318 outputs "1" when the 4th bit of the input value is "1" and the output of the AND gate 2316 is "1". Further, the AND gate 2319 outputs "1" when all of the outputs from the AND gates 2317 and 2318 are "1". Accordingly, when the output from the AND gate 2319 is "1", it becomes clear that the input value is "1624"～"1631".

[0113]

Likewise, the AND gates 2321～2326 decide that the input is "1632"～"1663". The AND gates 2327～2330 decide that the input is "1664"～"1791". The AND gates 2331～2333 decide that the input is "1792"～"2047". Further, the OR gates 2350～2356 decide that the input is "2048"～"524287" (values up to "524287" are decided because maxoverval is "325409").

Accordingly, by integrating these results of decisions by the OR gates 2336～2339, the comparator can decide that the input value is equal to or larger than "1624".

[0114]

As described above, while the comparator of the prior art apparatus must decide that the input is equal to or larger than L ×M, i.e., "1624", the comparator of this first embodiment decides that the input is equal to or larger than "1643", as can be seen in comparison between the 1st to 11th rows on table 2 and the 13th to 21st rows on table 2.

[0115]

Figure 12 shows the structure of the comparator in the overflow processing unit of the block interleaving apparatus according to the second embodiment of the invention.

In figure 12, 2321～2333 denote AND gates, and 2334, 2335, 2340～2342 denote OR gates.

In figure 12, the comparator should decide that the input is equal to or larger than "1643". However, since this decision is included in the decision of "1632" and larger values, this circuit decides that the input is equal to or larger than "1632".

[0116]

Initially, the AND gates 2321～2326 decide that the input is "1632"～"1663". The AND gates 2327～2330 decide that the input is "1664"～"1791". The AND gates 2331～2333 decide that the input is "1792"～"2047". Further, the OR gates 2334 and 2335 decide that the input is "2048"～"16383" (values up to "16383" are decided because maxoverval is "12967").

[0117]

Accordingly, by integrating these results of decisions by the OR gates 2340～2342, the comparator can decide that the input value is equal to or larger than "1632", i.e., "1643".

[0118]

The circuit shown in figure 12 requires thirteen AND gates and five OR gates while the prior art circuit shown in figure 11 requires twenty-two AND gates and six OR gates. That is, the circuit shown in figure 12 is reduced in circuit scale as compared with the prior art circuit because the objects to be compared are reduced, resulting in reduced area and reduced power consumption.

[0119]

By the way, the block interleaving with $L=8$, $M=203$, and $\alpha=20$ is effectively used for error correction in BS digital broadcasting.

To be specific, in the case of BS digital broadcasting, one data segment to be a target of correction by a Reed-Solomon decoder has 203 bytes in a data interleaving apparatus and, if the number of columns in a block interleaving apparatus at the transmitting end is 203, the correction ability of the Reed-Solomon decoder can be improved with the least storage capacity of the interleaving apparatus. Further, as the numbers of rows and columns are increased, the correction ability of the Reed-Solomon decoder against continuous burst errors is improved.

[0120]

Accordingly, in a block deinterleaving apparatus at the corresponding receiving end, by setting $L=8$, $M=203$, and $\alpha=20$ as described above, it is possible to provide a block deinterleaving apparatus which can improve the correction ability against the burst errors with the minimum circuit scale. Further, α may be an arbitrary integer not less than 2 so long as α has no common divisor with $L\times M-1$ and is not equal to $M^{(-x)}$, but the greatest effect is obtained when α is 20.

[0121]

As described above, the block deinterleaving apparatus according to the second embodiment of the invention is provided with the $L\times M$ data storage unit which generates an output from the block deinterleaving apparatus, the address generation unit which outputs addresses to the storage unit, and the storage unit control unit which outputs a control signal to the storage unit. In the address generation unit, the 0th address Ab(0) of a block having a block number b is set at 0, and the n-th (n: integer, $0 \leq n$) address Ab(n) of this block is generated from the remainder which is left when dividing the sum of the product of α (α: integer, $2 \leq \alpha$) and $L^{(b-x)}$ (x: integer, $0 \leq x \leq b$) and Ab(n-1) by $L \times M-1$, and reading and writing are repeated from/in the address so generated, thereby performing block deinterleaving. Therefore, the storage unit and the address generation unit are optimized, and the block deinterleaving can be performed with the minimum circuit scale.

[0122]

Further, since the first address and the last address of each block are constant, two pieces of data in these addresses can be processed simultaneously by allocating a continuous area of the storage unit to these addresses, whereby the number of accesses to the storage unit is reduced, resulting in reduction in power consumption of the address generation unit.

[0123]

Further, especially when performing block deinterleaving with L=8 and M=203, in the prior art address generation unit disclosed in Japanese Published Patent Application No. Hei.8-511393, the 0th address Ab(0) of a block having a block number b is set at 0, and the n-th (n: integer, $0 \leq n$) address Ab(n) of this block is generated from the remainder which is left when dividing the sum of the product of $L^{(b-x)}$ (x: integer, $0 \leq x \leq b$) and Ab(n-1) by L×M-1. In repetition of this calculation, the value to be divided by L×M-1 increases infinitely. So, when implementing this calculation with a circuit, the circuit is composed of a multiplier which sets the initial value at $L^{(b-x-1)}$, multiplies the input by L, and outputs the product to an overflow processing unit 1 (hereinafter referred to as a remainder generator 1); the remainder generator 1 which outputs the remainder obtained by dividing the input by L×M-1, to the multiplier and an adder; the adder which adds Ab(n-1) to the output from the remainder generator 1 and outputs the sum to an overflow processing unit 2 (hereinafter, referred to as a remainder generator 2); and the remainder generator 2 which generates Ab(n) as the remainder obtained by dividing the input by L×M-1. The remainder generator 1 is composed of a comparator and a subtracter for subtracting L×M-1 from the input until the input becomes equal to or lower than L×M-1. In this case, since the minimum value to be subjected to the subtraction is "1624", the comparator is required of the function of deciding "1624" and larger values.

[0124]

However, in the block deinterleaving apparatus according to the second embodiment, assuming that $\alpha=20$, L=8, and M=203, the circuit is composed of a multiplier which sets the initial value at the product of $L^{(b-x-1)}$ and α, multiplies the input by L, and outputs the product to a remainder generator 1; the remainder generator 1 which outputs the remainder obtained by dividing the input by L×M-1, to the multiplier and an adder; the adder which adds Ab(n-1) and the output from the remainder generator 1 and outputs the sum to a remainder generator 2; and the remainder generator 2 which generates Ab(n) as the remainder obtained by dividing the input by L×M-1. The remainder generator 1 is composed of a comparator and a subtracter for subtracting L×M-1 from the input until the input becomes equal to or lower than L×M-1. In this case, since the minimum value to be subjected to the subtraction is "1643", the comparator is required of the function of deciding "1643" and larger values. Therefore, as compared with the prior art circuit, the area of the comparator is reduced, and the block deinterleaving can be performed with the minimum circuit area.

[0125]

Further, it is also possible to realize block deinterleaving by setting the reading address and the writing address at Ab(n) and Ab(n-t) (t: natural number, $t \leq L \times M-2$), respectively, and repeating reading and writing from/in each address at each point of time.

Furthermore, when Ab(0) is set at β (β: natural number, $\beta \leq L \times M-1$), an address Ab(n) may be generated from the remainder which is left when dividing the sum of the product of α and $M^{(b-x)}$

and Ab(n-1) by L×M-1.

[0126]

[Effects of the invention]

As described above, a block interleaving apparatus according to the invention of Claim 1 performs writing while skipping addresses at the same interval corresponding to at least one address into a storage unit to which the addresses are sequentially assigned, and performs reading in such order that the read data become block-interleaved data to obtain output data, and thereafter, performs writing in the same order as the reading, thereby performing block interleaving. Therefore, a block interleaving operation on a single plane of a storage unit having a storage area of one block can be carried out, and the circuit scale of a storage control unit can be reduced.

[0127]

Further, according to the invention of Claim 2, the block interleaving apparatus defied in Claim 1 comprises a storage unit for L×M pieces of data for generating an output of the block interleaving apparatus; an address generation unit for generating addresses to be output to the storage unit; and a storage control unit for outputting a control signal to the storage unit; wherein, in said address generation unit, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of

Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $M^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving. Therefore, the storage unit and the address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0128]

Further, according to the invention of Claim 3, in the block interleaving apparatus defined in Claim 2, the values of α and L ×M-1 are determined so that no common divisor exists between them. Therefore, the address generation rule is prevented from being failed, and the storage unit and the address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0129]

Further, according to the invention of Claim 4, in the block interleaving apparatus defined in Claim 2, the values of α and $M^{(-x)}$ are determined so that α is not equal to $M^{(-x)}$. Therefore, it is prevented that writing is carried out into continuous addresses at the initial writing, and the storage unit and the address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0130]

Further, according to the invention of Claim 5, in the block interleaving apparatus defined in Claim 2, the values of α, L, and M are determined at 20, 8, and 203, respectively. Therefore, the area of a comparator as a component of the address generation unit is reduced, and the storage unit and the address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0131]

Further, according to the invention of Claim 6, a block deinterleaving apparatus performs writing while skipping addresses at the same interval corresponding to at least one address into a storage unit to which the addresses are sequentially assigned, and performs reading in such order that the read data become block-deinterleaved data to obtain output data, and thereafter, performs writing in the same order as the reading, thereby performing block deinterleaving. Therefore, a block deinterleaving operation on a single plane of a storage unit having a storage area of one block can be carried out, and the circuit scale of a storage control unit can be reduced.

[0132]

Further, according to the invention of Claim 7, the block deinterleaving apparatus defined in Claim 6 comprises a storage unit for L×M pieces of data for generating an output of the block deinterleaving apparatus; an address generation unit for generating addresses to be output to the storage unit; and a storage control unit for outputting a control signal to the storage unit; wherein, in said address generation unit, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $L^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving. Therefore, the storage unit and the address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0133]

Further, according to the invention of Claim 8, in the block deinterleaving apparatus defined in Claim 7, the values of α and L×M-1 are determined so that no common divisor exists between them. Therefore, the address generation rule is prevented from being failed, and the storage unit and the address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0134]

Further, according to the invention of Claim 9, in the block deinterleaving apparatus defined in Claim 7, the values of α and $L^{(-x)}$ are determined so that α is not equal to $L^{(-x)}$. Therefore, it is prevented that writing is carried out into continuous addresses at the initial writing, and the storage unit and the address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0135]

Further, according to the invention of Claim 10, in the block deinterleaving apparatus defined in Claim 7, the values of α, L, and M are determined at 20, 8, and 203, respectively. Therefore, the area of a comparator as a component of the address generation unit is reduced, and the storage unit and the address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0136]

Further, according to the invention of Claim 11, a block interleaving method includes performing writing while skipping addresses at the same interval corresponding to at least one address into a storage unit to which the addresses are sequentially assigned; performing reading in such order that the read data become block-interleaved data to obtain output data; and thereafter performing writing in the same order as the reading, thereby performing block interleaving. Therefore, a block interleaving operation on a single plane of a storage unit having a storage area of one block can be carried out, and the circuit scale of a storage control unit can be reduced.

[0137]

According to the invention of Claim 12, in the block interleaving method defined in Claim 11, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $M^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving. Therefore, a storage unit and an address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0138]

Further, according to the invention of Claim 13, in the block interleaving method defined in Claim 12, the values of α and L×M-1 are determined so that no common divisor exists between them. Therefore, the address generation rule is prevented from being failed, and a storage unit and an address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0139]

According to the invention of Claim 14, in the block interleaving method defined in Claim 12, the values of α and $M^{(-x)}$ are determined so that α is not equal to $M^{(-x)}$. Therefore, it is prevented that writing is carried out into continuous addresses at the initial writing, and a storage unit and an address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0140]

According to the invention of Claim 15, in the block interleaving method defined in Claim 12, the values of α, L, and M are determined at 20, 8, and 203, respectively. Therefore, the area of a comparator as a component of an address generation unit is reduced, and a storage unit and the address generation unit are optimized, whereby block interleaving can be carried out with a minimum circuit area.

[0141]

Further, according to the invention of Claim 16, a block deinterleaving method includes performing writing while skipping addresses at the same interval corresponding to at least one address, into a storage unit to which the addresses are sequentially assigned; performing reading in such order that the read data become block-deinterleaved data to obtain output data; and thereafter, performing writing in the same order as the reading, thereby performing block deinterleaving. Therefore, a block deinterleaving operation on a single plane of a storage unit having a storage area of one block can be carried out, and the circuit scale of a storage control unit can be reduced.

[0142]

Further, according to the invention of Claim 17, in the block deinterleaving method according to Claim 16, 0th address Ab(0) of block number b (b: integer not less than 0) is 0, and n-th (n: integer not less than 0) address Ab(n) of the block number b is generated from a remainder obtained in dividing a result of addition of Ab(n-1) to a result of multiplication of α (α: integer not less than 2) and $L^{(b-x)}$ (x: integer not less than 0 and not larger than b), by L×M-1; and reading and writing are repeated on the generated addresses to perform block interleaving. Therefore, a storage unit and an address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0143]

Further, according to the invention of Claim 18, in the block deinterleaving method defined in Claim 17, the values of α and L×M-1 are determined so that no common divisor exists between them. Therefore, the address generation rule is prevented from being failed, and a storage unit and an address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0144]

Further, according to the invention of Claim 19, in the block deinterleaving method defined in Claim 17, the values of α and $L^{(-x)}$ are determined so that α is not equal to $L^{(-x)}$. Therefore, it is prevented that writing is carried out into continuous addresses at the initial writing, and a storage unit and an address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[0145]

Further, according to the invention of Claim 20, in a block deinterleaving method defined in Claim 17, the values of α, L, and M are determined at 20, 8, and 203, respectively. Therefore, the area of a comparator as a component of an address generation unit is reduced, and a storage unit and the address generation unit are optimized, whereby block deinterleaving can be carried out with a minimum circuit area.

[Brief Description of the Drawings]

[Figure 1] A block diagram illustrating the structure of a block interleaving apparatus according to the first embodiment of the present invention.

[Figure 2] A diagram for explaining an example of data writing/reading order in/from a storage unit in the block interleaving apparatus according to the first embodiment of the invention.

[Figure 3] A block diagram for explaining the reason why only one storage unit suffices for the block interleaving apparatus according to the first embodiment of the invention.

[Figure 4] A diagram illustrating the structure of an overflow processing unit in the block interleaving apparatus according to the first embodiment of the invention as well as the prior art.

[Figure 5] A diagram illustrating the structure of a comparator in a storage control unit in the prior art block interleaving apparatus.

[Figure 6] A diagram illustrating the structure of a comparator included in a storage control unit in the block interleaving apparatus according to the first embodiment of the invention.

[Figure 7] A block diagram illustrating the structure of a block deinterleaving apparatus according to a second embodiment of the invention.

[Figure 8] A diagram for explaining an example of data writing/reading order in/from a storage unit in the block deinterleaving apparatus according to the second embodiment of the invention.

[Figure 9] A block diagram for explaining the reason why only one storage unit suffices for the block deinterleaving apparatus according to the second embodiment of the invention.

[Figure 10] A diagram illustrating the structure of an overflow processing unit in the block deinterleaving apparatus according to the second embodiment of the invention as well as the prior art.

[Figure 11] A diagram illustrating the structure of a comparator in a storage control unit in the prior art block deinterleaving apparatus.

[Figure 12] A diagram illustrating the structure of a comparator in a storage control unit in the block deinterleaving apparatus according to the second embodiment of the invention.

[Figure 13] A diagram for explaining data writing/reading order in/from the storage units in the prior art block interleaving apparatus and block deinterleaving apparatus.

[Figure 14] A block diagram for explaining the reason why only one storage unit suffices for the prior art block interleaving apparatus and block deinterleaving apparatus.

[Description of Reference Numerals]

4, 104 ... storage unit

12, 112 ... control unit

3, 103 ... address generation unit

18, 19, 118, 119 ... constant generator

112, 116, 140, 141 ... overflow processing unit

[Name of the Document] Abstract

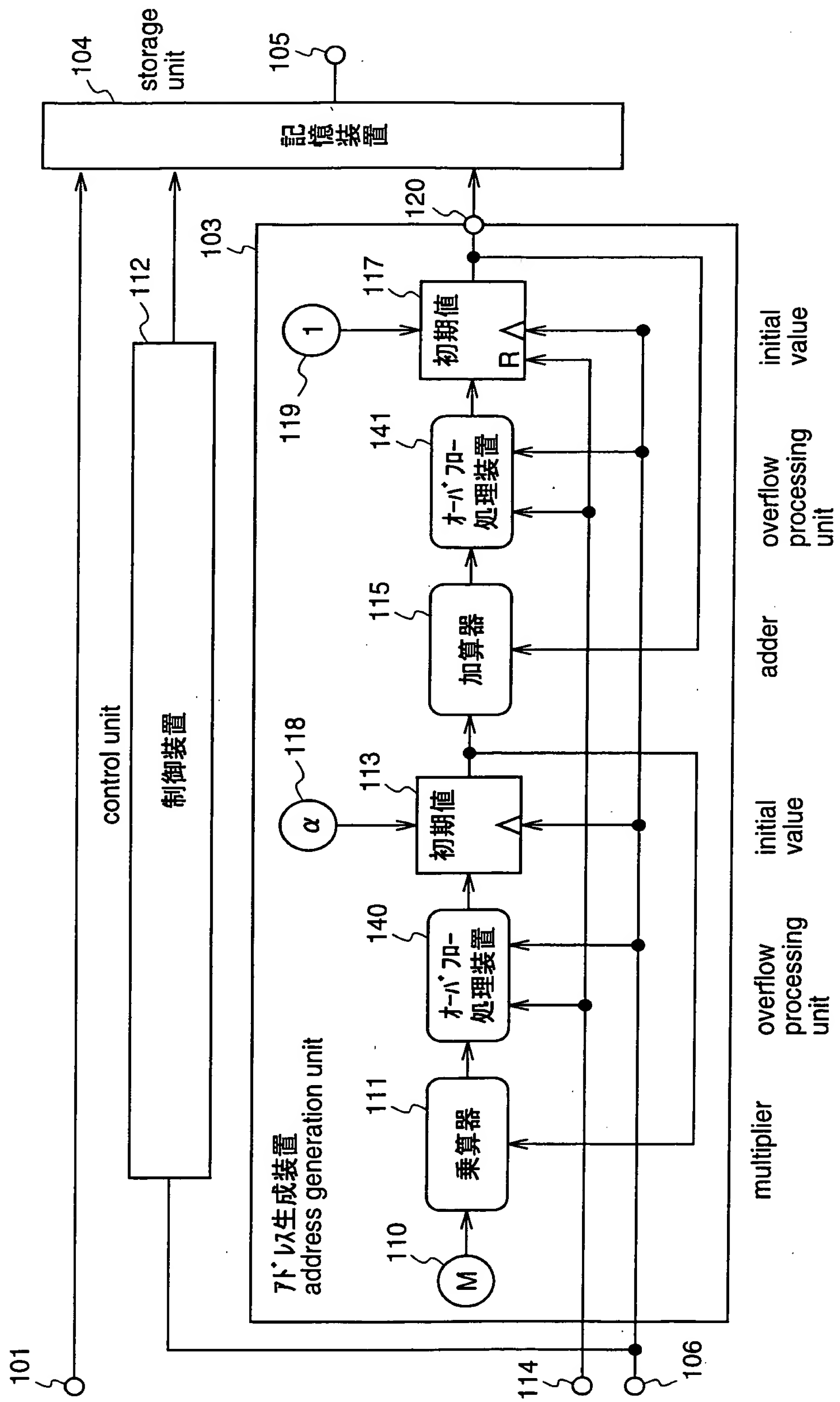
[Summary]

[Object] It is an object of the present invention to realize reduction in circuit area and low power consumption in a block interleaving apparatus, a block deinterleaving apparatus, a block interleaving method, and a block deinterleaving method, which perform block interleaving or block deinterleaving on a single plane of a storage unit having a storage area corresponding to one block.

[Solution] Writing is carried out so that at least one address is skipped at initial writing, by a control unit 112 for controlling a storage unit 104, and an address generation unit 103.

[Selected Figure] Fig.1

Name of Document
【書類名】　図面 Drawing
【図1】 Figure 1



【図2】 Figure 2

(a)

2

0	10	1	11	2
12	3	13	4	14
5	15	6	16	7
17	8	18	9	19

(b)

10

0	2	4	6	8
10	12	14	16	18
1	3	5	7	9
11	13	15	17	19

(c)

50-(19×2)=12

0	8	16	5	13
2	10	18	7	15
4	12	1	9	17
6	14	3	11	19

(d)

60-(19×3)=3

0	13	7	1	14
8	2	15	9	3
16	10	4	17	11
5	18	12	6	19

(e)

15

0	14	9	4	18
13	8	3	17	12
7	2	16	11	6
1	15	10	5	19

(f)

75-(19×3)=18

0	18	17	16	15
14	13	12	11	10
9	8	7	6	5
4	3	2	1	19

(g)

90-(19×4)=14

0	15	11	7	3
18	14	10	6	2
17	13	9	5	1
16	12	8	4	19

(h)

70-(19×3)=13

0	3	6	9	12
15	18	2	5	8
11	14	17	1	4
7	10	13	16	19

(i)

65-(19×3)=8

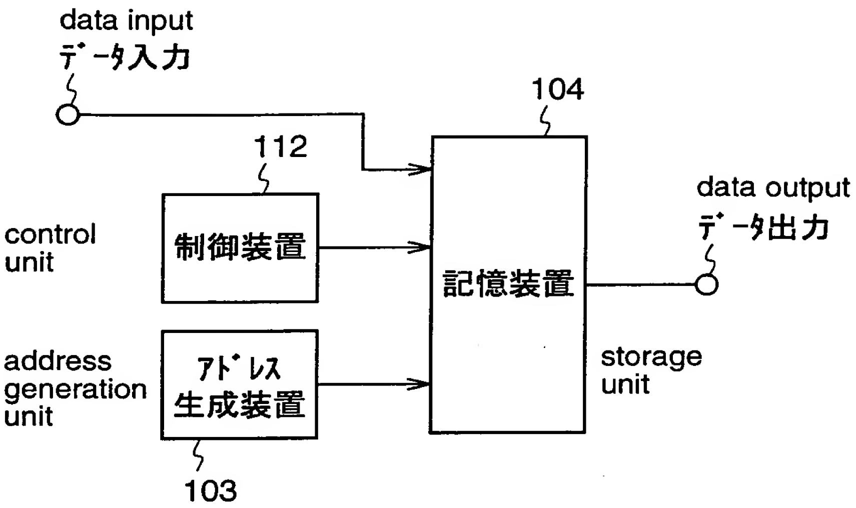
0	12	5	17	10
3	15	8	1	13
6	18	11	4	16
9	2	14	7	19

(j)

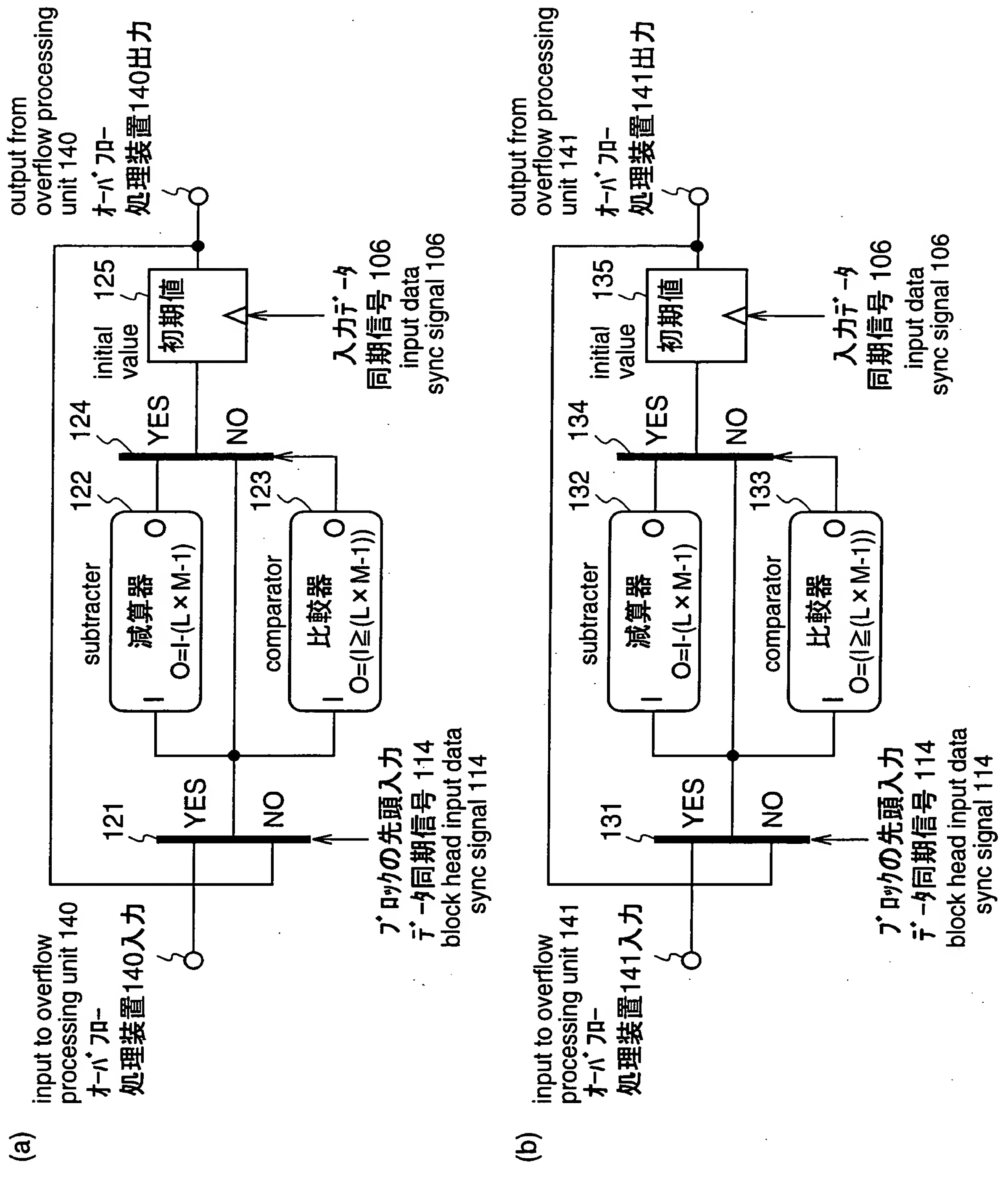
40-(19×2)=2

0	10	1	11	2
12	3	13	4	14
5	15	6	16	7
17	8	18	9	19

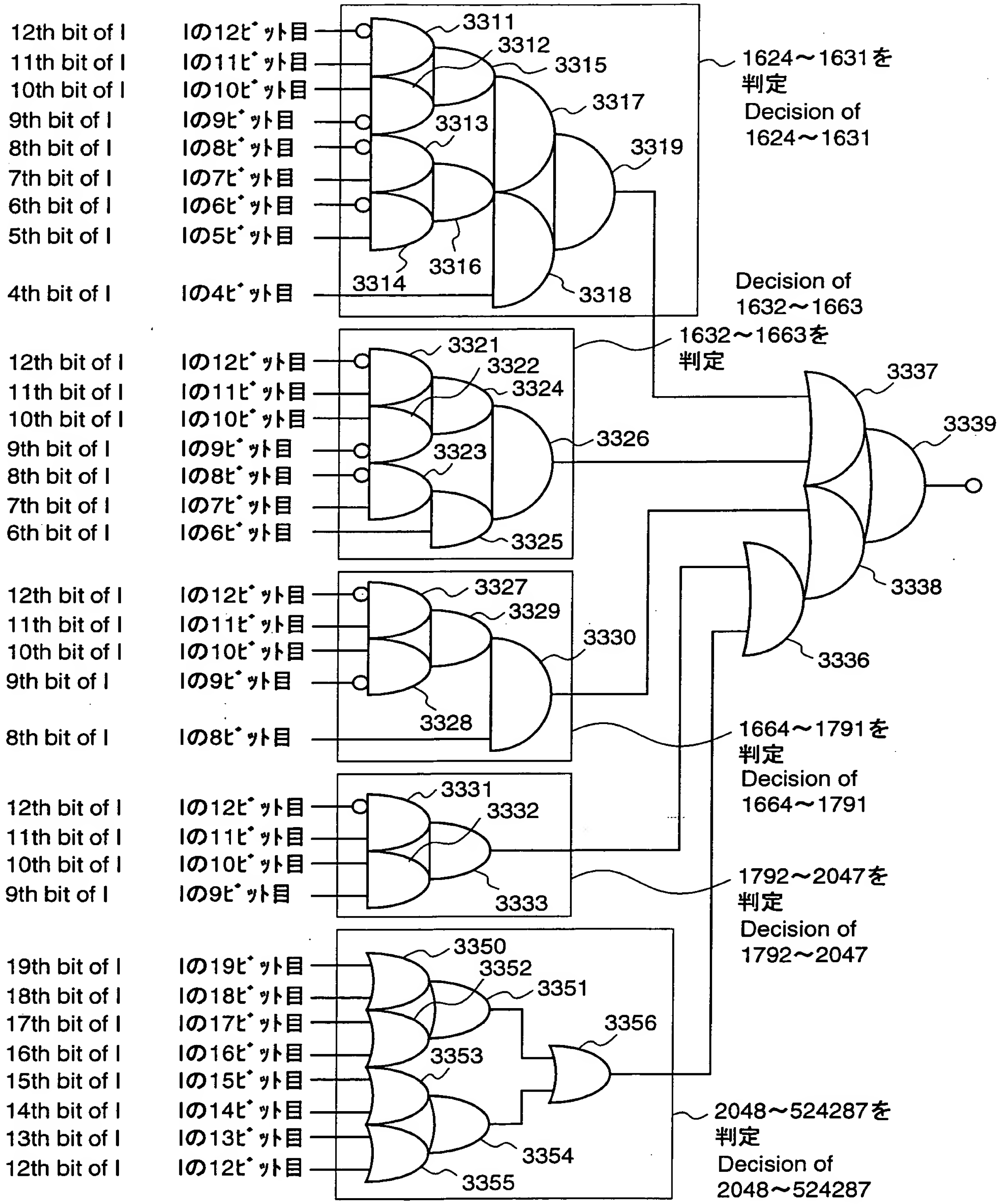
【図3】 Figure 3


data input
ﾃﾞｰﾀ入力
104
112
data output
ﾃﾞｰﾀ出力
control unit
制御装置
記憶装置
storage unit
address generation unit
ｱﾄﾞﾚｽ生成装置
103

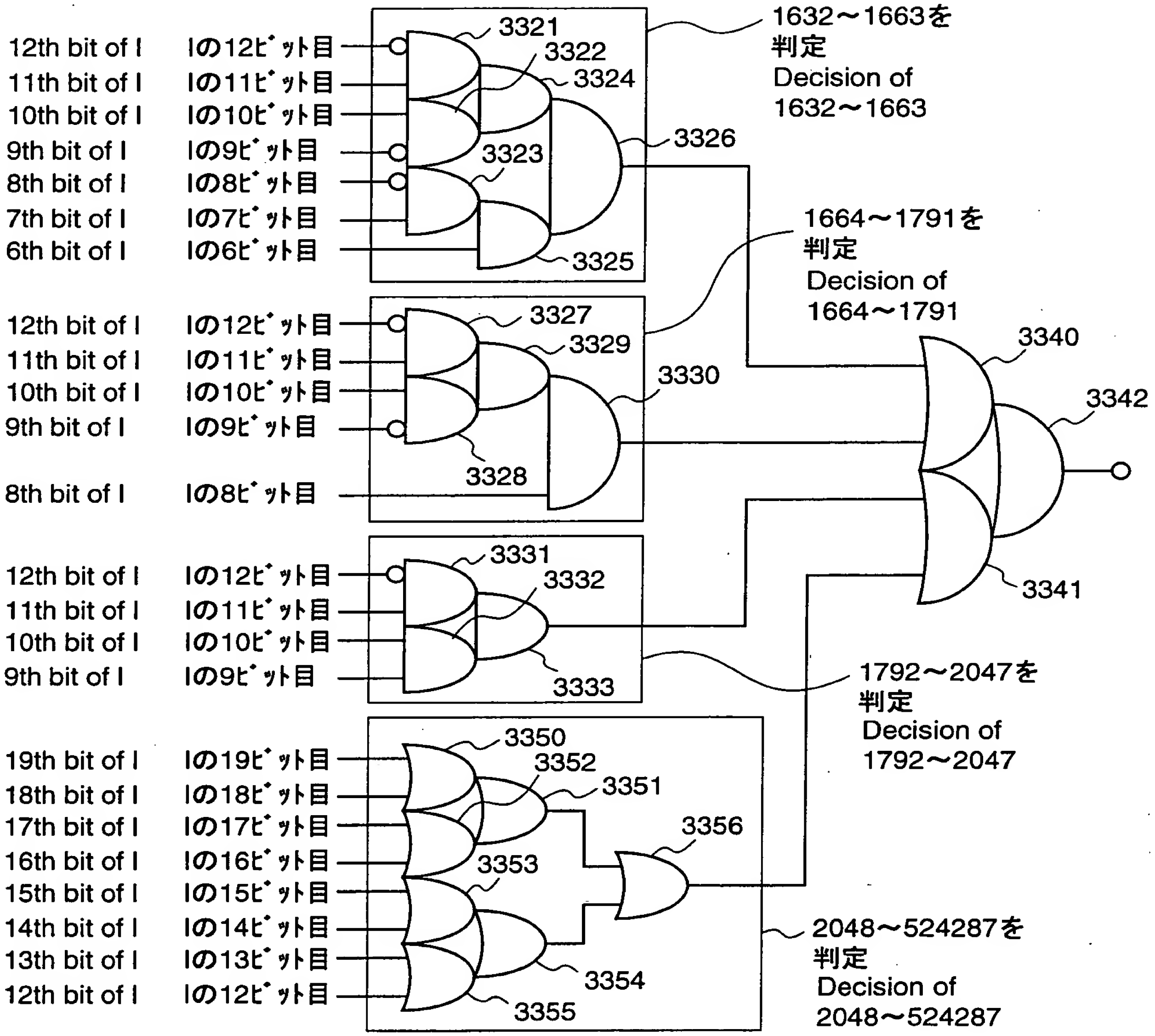
【図4】 Figure 4


(a)
input to overflow processing unit 140
オーバフロー処理装置140入力
121
YES
NO
ブロックの先頭入力データ同期信号 114
block head input data sync signal 114
subtracter
減算器
I
O
O=I-(L×M-1)
122
comparator
比較器
I
O
O=(I≧(L×M-1))
123
124
YES
NO
initial value
125
初期値
入力データ同期信号 106
input data sync signal 106
output from overflow processing unit 140
オーバフロー処理装置140出力
(b)
input to overflow processing unit 141
オーバフロー処理装置141入力
131
YES
NO
ブロックの先頭入力データ同期信号 114
block head input data sync signal 114
subtracter
減算器
I
O
O=I-(L×M-1)
132
comparator
比較器
I
O
O=(I≧(L×M-1))
133
134
YES
NO
initial value
135
初期値
入力データ同期信号 106
input data sync signal 106
output from overflow processing unit 141
オーバフロー処理装置141出力

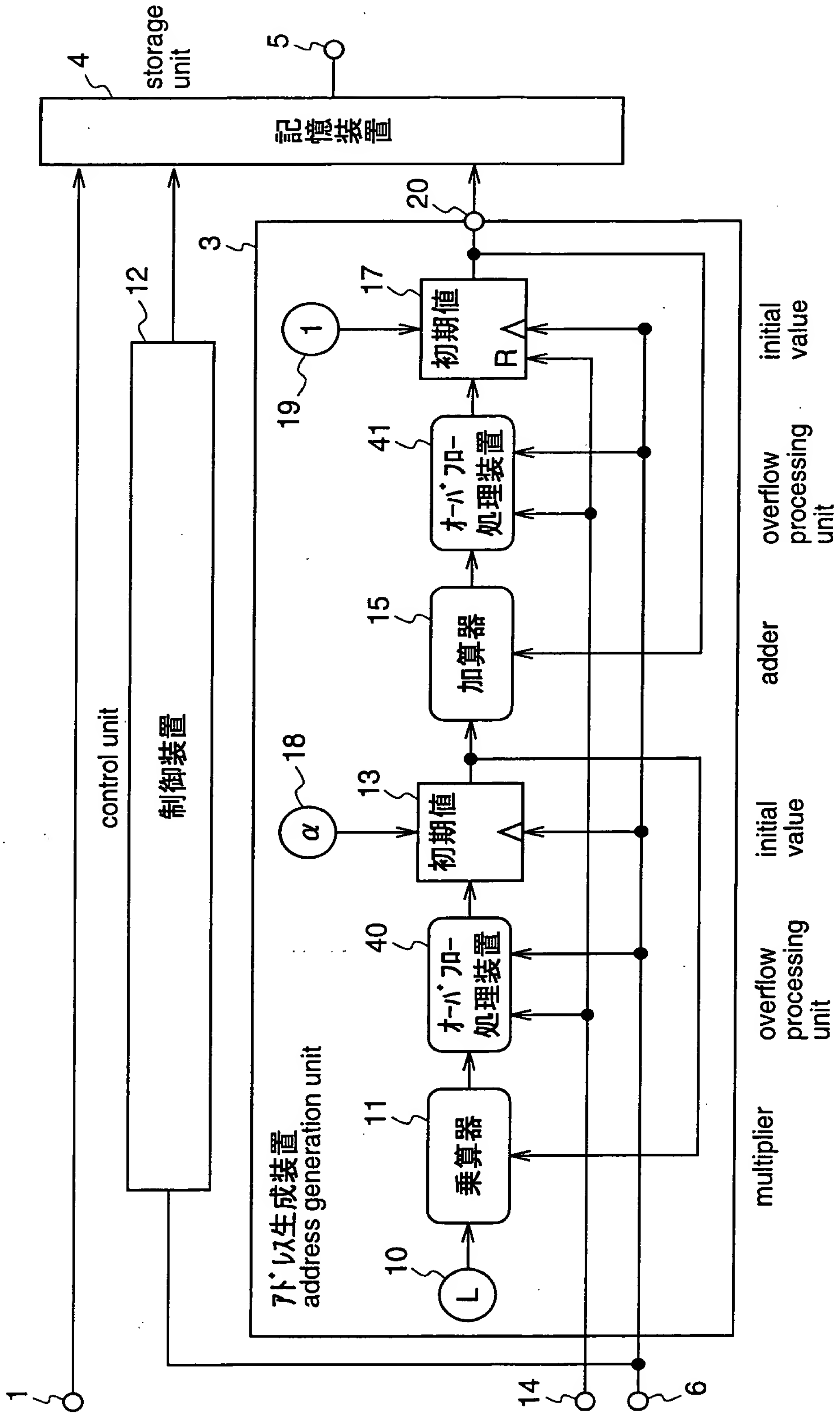
【図5】 Figure 5


12th bit of I
Iの12ﾋﾞｯﾄ目
11th bit of I
Iの11ﾋﾞｯﾄ目
10th bit of I
Iの10ﾋﾞｯﾄ目
9th bit of I
Iの9ﾋﾞｯﾄ目
8th bit of I
Iの8ﾋﾞｯﾄ目
7th bit of I
Iの7ﾋﾞｯﾄ目
6th bit of I
Iの6ﾋﾞｯﾄ目
5th bit of I
Iの5ﾋﾞｯﾄ目
4th bit of I
Iの4ﾋﾞｯﾄ目
3311
3312
3315
3317
3313
3319
3316
3314
3318
1624～1631を
判定
Decision of
1624～1631
Decision of
1632～1663
1632～1663を
判定
12th bit of I
Iの12ﾋﾞｯﾄ目
11th bit of I
Iの11ﾋﾞｯﾄ目
10th bit of I
Iの10ﾋﾞｯﾄ目
9th bit of I
Iの9ﾋﾞｯﾄ目
8th bit of I
Iの8ﾋﾞｯﾄ目
7th bit of I
Iの7ﾋﾞｯﾄ目
6th bit of I
Iの6ﾋﾞｯﾄ目
3321
3322
3324
3326
3323
3325
3337
3339
3338
12th bit of I
Iの12ﾋﾞｯﾄ目
11th bit of I
Iの11ﾋﾞｯﾄ目
10th bit of I
Iの10ﾋﾞｯﾄ目
9th bit of I
Iの9ﾋﾞｯﾄ目
8th bit of I
Iの8ﾋﾞｯﾄ目
3327
3329
3330
3328
3336
1664～1791を
判定
Decision of
1664～1791
12th bit of I
Iの12ﾋﾞｯﾄ目
11th bit of I
Iの11ﾋﾞｯﾄ目
10th bit of I
Iの10ﾋﾞｯﾄ目
9th bit of I
Iの9ﾋﾞｯﾄ目
3331
3332
3333
1792～2047を
判定
Decision of
1792～2047
19th bit of I
Iの19ﾋﾞｯﾄ目
18th bit of I
Iの18ﾋﾞｯﾄ目
17th bit of I
Iの17ﾋﾞｯﾄ目
16th bit of I
Iの16ﾋﾞｯﾄ目
15th bit of I
Iの15ﾋﾞｯﾄ目
14th bit of I
Iの14ﾋﾞｯﾄ目
13th bit of I
Iの13ﾋﾞｯﾄ目
12th bit of I
Iの12ﾋﾞｯﾄ目
3350
3352
3351
3356
3353
3354
3355
2048～524287を
判定
Decision of
2048～524287

【図6】 Figure 6


12th bit of I
11th bit of I
10th bit of I
9th bit of I
8th bit of I
7th bit of I
6th bit of I
Iの12ビット目
Iの11ビット目
Iの10ビット目
Iの9ビット目
Iの8ビット目
Iの7ビット目
Iの6ビット目
3321
3322
3324
3323
3325
3326
1632～1663を
判定
Decision of
1632～1663
12th bit of I
11th bit of I
10th bit of I
9th bit of I
8th bit of I
Iの12ビット目
Iの11ビット目
Iの10ビット目
Iの9ビット目
Iの8ビット目
3327
3329
3328
3330
1664～1791を
判定
Decision of
1664～1791
3340
3342
3341
12th bit of I
11th bit of I
10th bit of I
9th bit of I
Iの12ビット目
Iの11ビット目
Iの10ビット目
Iの9ビット目
3331
3332
3333
1792～2047を
判定
Decision of
1792～2047
19th bit of I
18th bit of I
17th bit of I
16th bit of I
15th bit of I
14th bit of I
13th bit of I
12th bit of I
Iの19ビット目
Iの18ビット目
Iの17ビット目
Iの16ビット目
Iの15ビット目
Iの14ビット目
Iの13ビット目
Iの12ビット目
3350
3352
3351
3353
3356
3354
3355
2048～524287を
判定
Decision of
2048～524287

【図7】 Figure 7


4
storage unit
記憶装置
5
20
3
17
初期値
R
1
19
41
ｵｰﾊﾞﾌﾛｰ処理装置
15
加算器
18
α
13
初期値
40
ｵｰﾊﾞﾌﾛｰ処理装置
11
乗算器
10
L
ｱﾄﾞﾚｽ生成装置
address generation unit
12
制御装置
control unit
initial value
overflow processing unit
adder
initial value
overflow processing unit
multiplier
1
14
6

【図8】 Figure 8

(a)

2

0	10	1	11	2
12	3	13	4	14
5	15	6	16	7
17	8	18	9	19

(b)

8

0	12	5	17	10
3	15	8	1	13
6	18	11	4	16
9	2	14	7	19

(c)

32-19=13

0	3	6	9	12
15	18	2	5	8
11	14	17	1	4
7	10	13	16	19

(d)

52-(19×2)=14

0	15	11	7	3
18	14	10	6	2
17	13	9	5	1
16	12	8	4	19

(e)

56-(19×2)=18

0	18	17	16	15
14	13	12	11	10
9	8	7	6	5
4	3	2	1	19

(f)

72-(19×3)=15

0	14	9	4	18
13	8	3	17	12
7	2	16	11	6
1	15	10	5	19

(g)

60-(19×3)=3

0	13	7	1	14
8	2	15	9	3
16	10	4	17	11
5	18	12	6	19

(h)

12

0	8	16	5	13
2	10	18	7	15
4	12	1	9	17
6	14	3	11	19

(i)

48-(19×2)=10

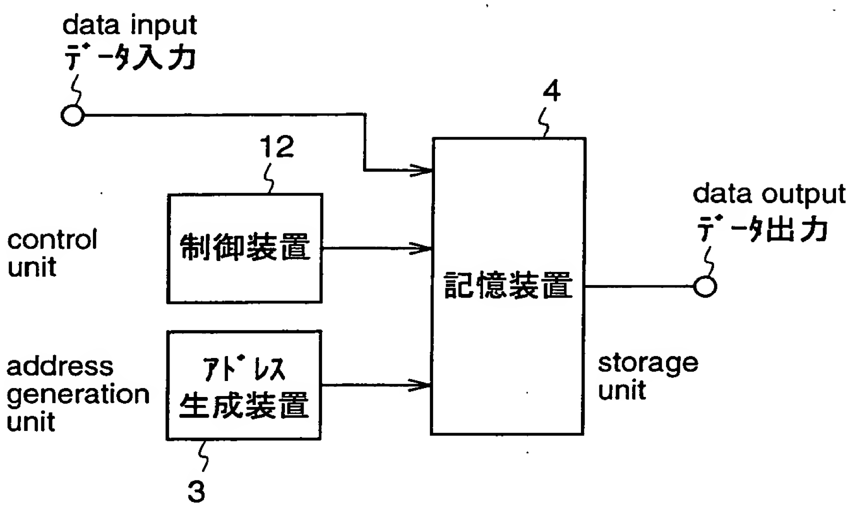
0	2	4	6	8
10	12	14	16	18
1	3	5	7	9
11	13	15	17	19

(j)

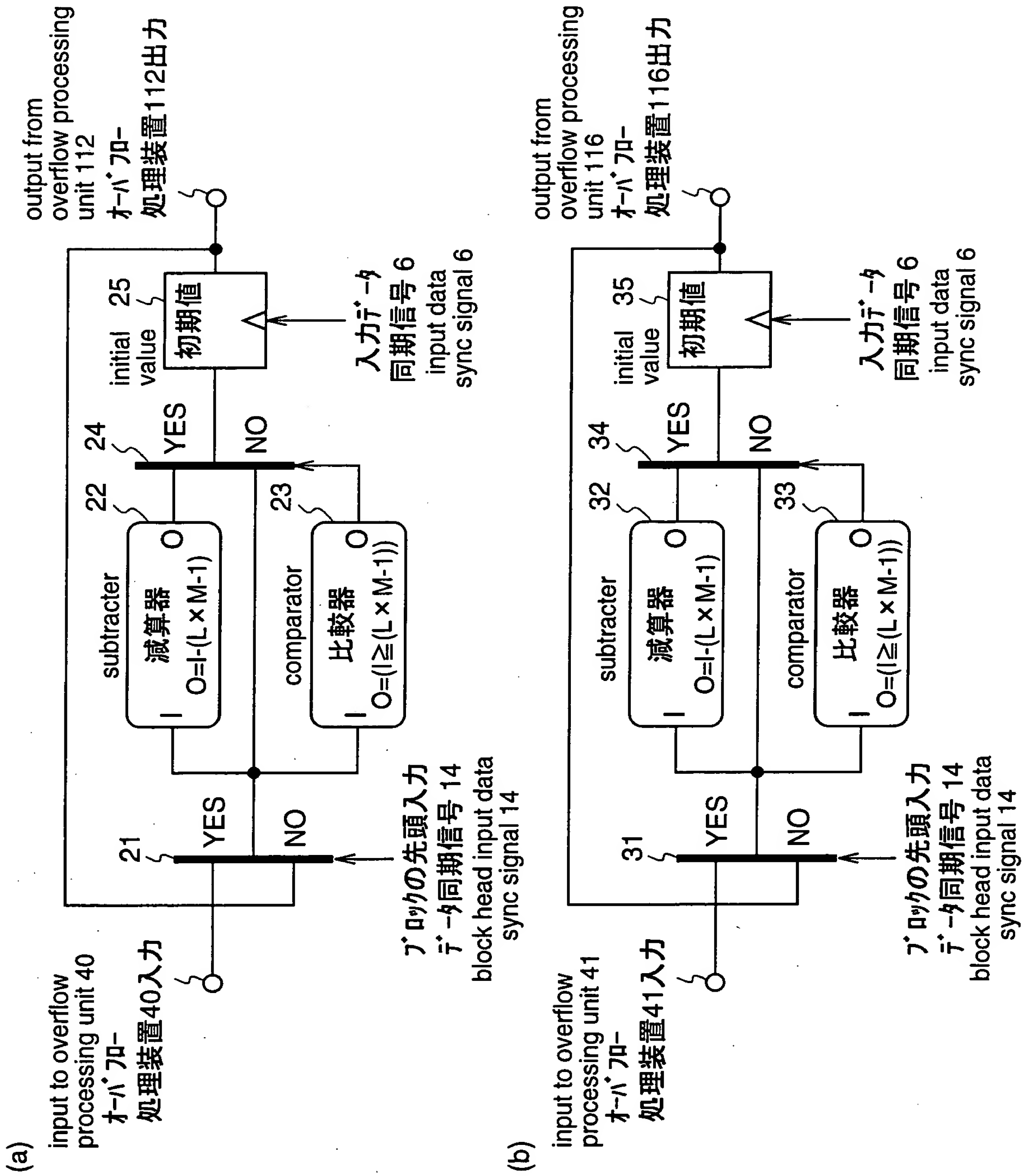
40-(19×2)=2

0	10	1	11	2
12	3	13	4	14
5	15	6	16	7
17	8	18	9	19

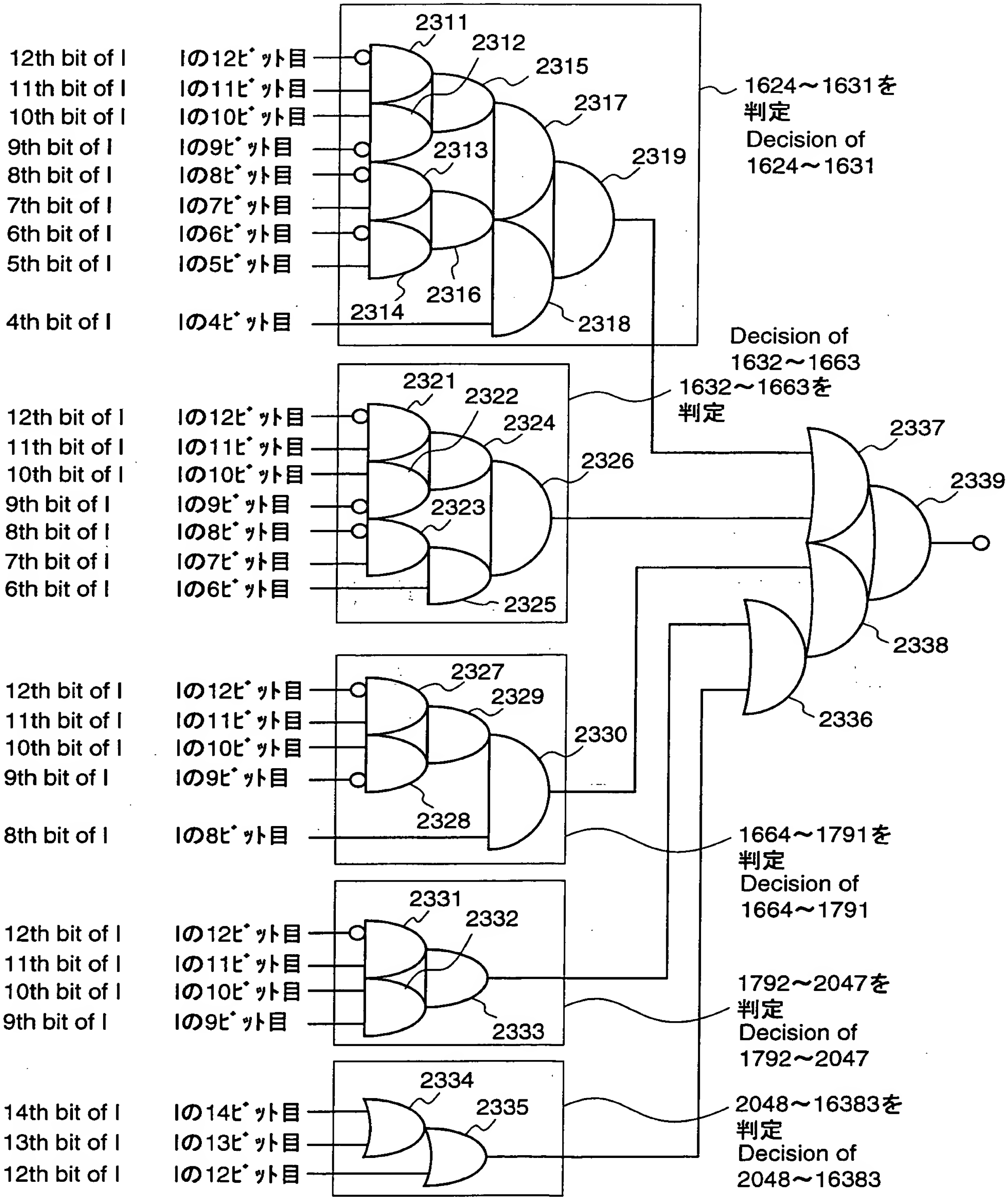
【図9】 Figure 9


data input
データ入力
4
12
制御装置
control unit
data output
データ出力
記憶装置
storage unit
address generation unit
アドレス
生成装置
3

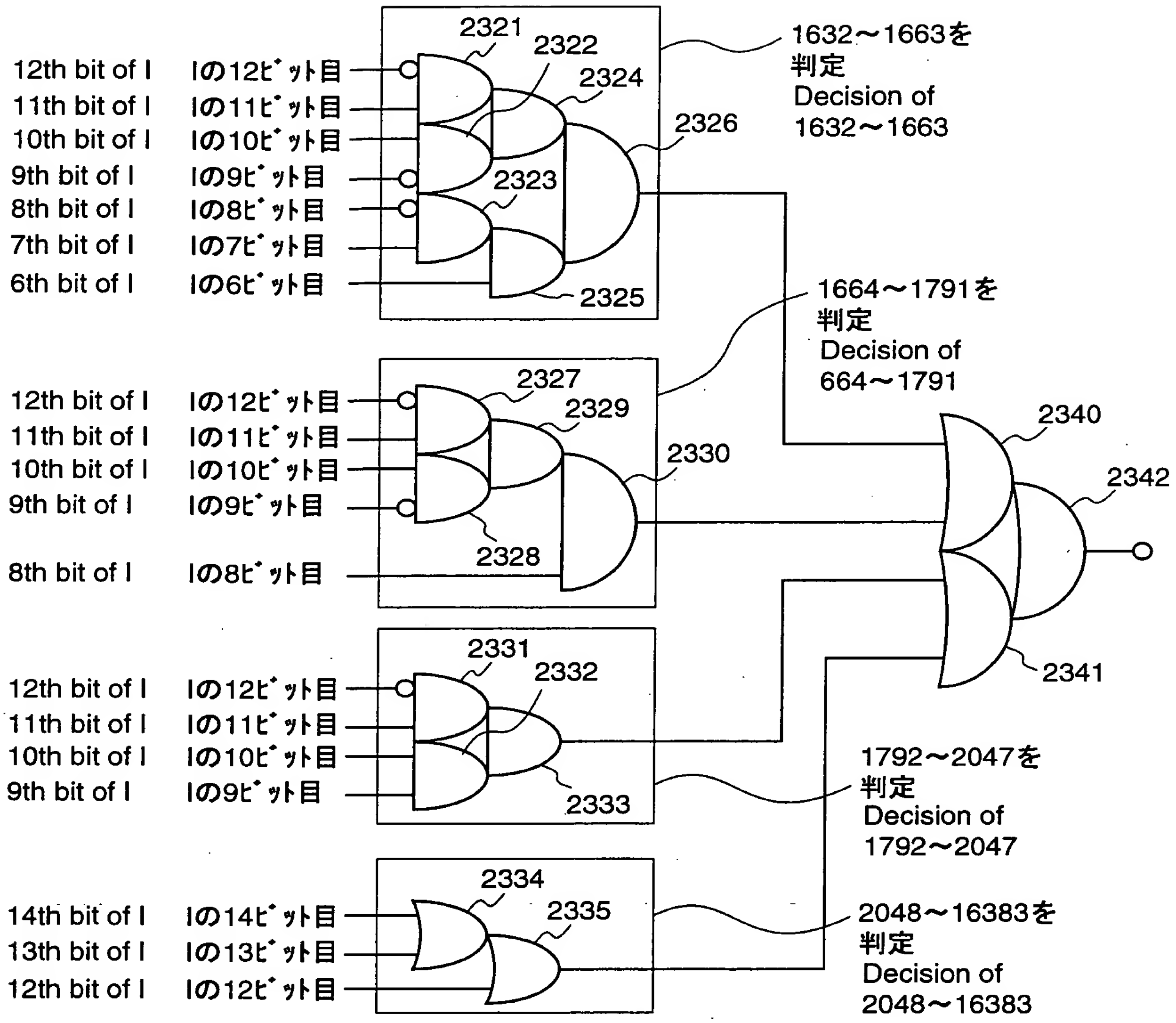
【図10】 Figure 10



【図11】 Figure 11



【図12】 Figure 12



【図13】 Figure 13

(a) 1

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

(b) 5

0	4	8	12	16
1	5	9	13	17
2	6	10	14	18
3	7	11	15	19

(c) 25-19=6

0	16	13	10	7
4	1	17	14	11
8	5	2	18	15
12	9	6	3	19

(d) 30-19=11

0	7	14	2	9
16	4	11	18	6
13	1	8	15	3
10	17	5	12	19

(e) 55-(19×2)=17

0	9	18	8	17
7	16	6	15	5
14	4	13	3	12
2	11	1	10	19

(f) 85-(19×4)=9

0	17	15	13	11
9	7	5	3	1
18	16	14	12	10
8	6	4	2	19

(g) 45-(19×2)=7

0	11	3	14	6
17	9	1	12	4
15	7	18	10	2
13	5	16	8	19

(h) 35-19=16

0	6	12	18	5
11	17	4	10	16
3	9	15	2	8
14	1	7	13	19

(i) 80-(19×4)=4

0	5	10	15	1
6	11	16	2	7
12	17	3	8	13
18	4	9	14	19

(j) 20-19=1

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

(k) 1

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

(l) 4

0	5	10	15	1
6	11	16	2	7
12	17	3	8	13
18	4	9	14	19

(m) 16

0	6	12	18	5
11	17	4	10	16
3	9	15	2	8
14	1	7	13	19

(n) 64-(19×3)=7

0	11	3	14	6
17	9	1	12	4
15	7	18	10	2
13	5	16	8	19

(o) 28-19=9

0	17	15	13	11
9	7	5	3	1
18	16	14	12	10
8	6	4	2	19

(p) 36-19=17

0	9	18	8	17
7	16	6	15	5
14	4	13	3	12
2	11	1	10	19

(q) 68-(19×3)=11

0	7	14	2	9
16	4	11	18	6
13	1	8	15	3
10	17	5	12	19

(r) 44-(19×2)=6

0	16	13	10	7
4	1	17	14	11
8	5	2	18	15
12	9	6	3	19

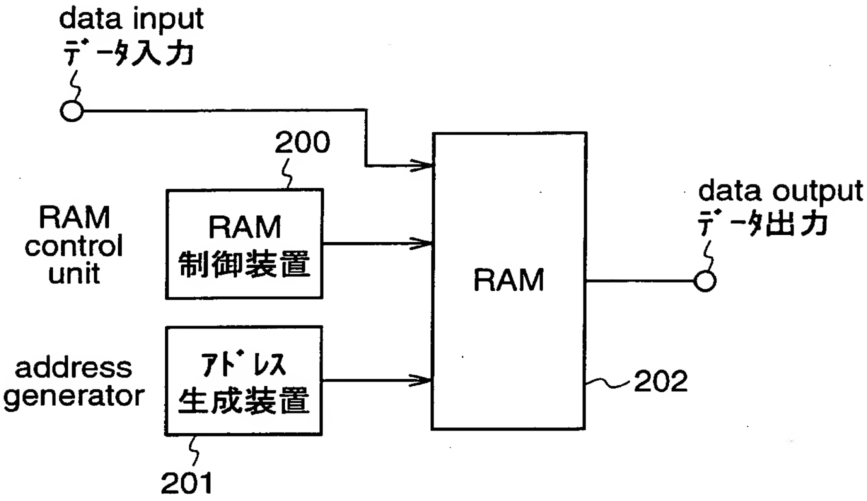
(s) 24-19=5

0	4	8	12	16
1	5	9	13	17
2	6	10	14	18
3	7	11	15	19

(t) 20-19=1

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

【図14】 Figure 14


data input
ﾃﾞｰﾀ入力
200
RAM control unit
RAM
制御装置
address generator
ｱﾄﾞﾚｽ
生成装置
201
RAM
202
data output
ﾃﾞｰﾀ出力